

DIGITAL COMPUTER

LIBRARY



LIBRARY
OF THE
UNIVERSITY
OF ILLINOIS

NOTICE: According to Sec. 19
(a) of the *University Statutes*,
all books and other library
materials acquired in any man-
ner by the University belong to
the University Library. When
this item is no longer needed
by the department, it should
be returned to the Acquisition
Department, University Library.

COMPUTER
SCIENCE



Digitized by the Internet Archive
in 2026 with funding from
University of Illinois Urbana-Champaign



Proceedings

SIXTH ANNUAL ALLERTON CONFERENCE ON CIRCUIT AND SYSTEMS THEORY

October 2-4, 1968

University of Illinois

Allerton House, Monticello, Illinois

Sponsored by the

Department of Electrical Engineering and the

Coordinated Science Laboratory of the

University of Illinois, Urbana, Illinois and the

Circuit Theory and Automatic Control Groups of the

Institute of Electrical & Electronics Engineers

IEEE Catalog No. 68 C 44 CT

PROCEEDINGS
SIXTH ANNUAL ALLERTON CONFERENCE
ON CIRCUIT AND SYSTEM THEORY

EDITED BY

T. N. TRICK
R. T. CHIEN

Conference Co-Chairmen

Conference held

October 2, 3, 4, 1968
Allerton House
Monticello, Illinois

Sponsored by the

DEPARTMENT OF ELECTRICAL ENGINEERING
and the
COORDINATED SCIENCE LABORATORY
of the
UNIVERSITY OF ILLINOIS
Urbana, Illinois

and the

CIRCUIT THEORY AND AUTOMATIC CONTROL GROUPS
of the
INSTITUTE OF ELECTRICAL & ELECTRONICS ENGINEERS

DIGITAL COMPUTER
LABORATORY
LIBRARY

FOREWORD

The Sixth Annual Allerton Conference on Circuit and System Theory was held at Allerton House from October 2-4, 1968. These Proceedings are a permanent record of the papers presented at that Conference. The papers describe new results in information and coding theory, digital systems, communication sciences, control systems, network theory, and computer-aided design.

In addition to the technical sessions, the remoteness of Allerton Park continues to provide a setting fruitful to the exchange of ideas throughout the Conference.

The Conference was co-sponsored by the Department of Electrical Engineering and the Coordinated Science Laboratory of the University of Illinois, and the IEEE Groups on Circuit Theory and Automatic Control. We acknowledge the encouragement and support of Professor W. Dale Compton, Director of the Coordinated Science Laboratory, and Professor E. C. Jordan, Head of the Department of Electrical Engineering at the University of Illinois. Special thanks are due Mrs. Divona Keel, who coordinated with extraordinary skill all Conference arrangements. Her assistance was indispensable in making the Conference successful. We thank Mr. Harold B. Lawler for his help with certain business aspects of the Conference. We are grateful to the secretaries, Mrs. Rosa Townsend, Angie Johnson, Rose Lane, and Sherry Kallembach, for their assistance. Also, we thank K. S. Fu for organizing the special session on pattern recognition. In addition, we are indebted to our colleagues for their vital assistance: Jose B. Cruz, Jr., Abraham Haddad, Petar V. Kokotović, Wataru Mayeda, Juraj Medanić, Gernot Metze, William R. Perkins, Franco Preparata, W. J. Poppelbaum, Jiri Vlach and Nelson Wax.

R. T. Chien
T. N. Trick

TABLE OF CONTENTS

Foreword	iii
GENERAL SESSION A	
Chairman: R. T. Chien, University of Illinois	
NEW DIRECTIONS AND HORIZONS IN CIRCUIT AND SYSTEM THEORY . . .	
M. E. Van Valkenburg, Princeton University	
WHAT A SYSTEM THEORIST SHOULD KNOW ABOUT CODING!	
James L. Massey, University of Notre Dame	
SPECIAL SESSION -- PATTERN RECOGNITION-THEORY & PRACTICE	1
Organizer: K. S. Fu, Purdue University	
Chairman: C. K. Chow, IBM	
ALGORITHMS FOR PATTERN CLASSIFICATION	1
R. L. Kashyap, Purdue University	
ON FEATURE SELECTION IN PATTERN RECOGNITION	10
K. S. Fu, T. J. Li, Purdue University and P. J. Min, IBM	
ON THE APPLICATIONS OF PATTERN RECOGNITION TO GEOPHYSICAL AND GEOLOGICAL PROBLEMS	20
Z. J. Nikolic, Esso Production Research Company	
COMPUTER-ASSISTED FINGERPRINT ENCODING AND IDENTIFICATION . .	30
C. N. Liu, IBM and Purdue University	
PATTERN CLASSIFICATION OF DATA USING A REINFORCEMENT LEARNING SCHEME	31
R. W. McLaren, G. C. Breshears, and A. C. Pan, University of Missouri, Columbia	
APPLICATIONS OF OPTIMAL CONTROL	42
Chairman: P. Kokotovic, M. Pupin Institute, Belgrade and University of Illinois	
APPLICATION OF OPTIMAL CONTROL TO SOME HIGH SPEED GROUND TRANSPORT PROBLEMS	42
Michael Athans, Massachusetts Institute of Technology	
AN OPTIMAL CONTROL ALGORITHM FOR RAMP METERING OF URBAN FREEWAYS	52
Li Shin Yuan and John B. Kreer, Michigan State University	
A ROUTING ANALOGY IN THE CONTROL SPACE	62
William G. Vogt and Marlin H. Mickle, University of Pittsburgh	

ON OPTIMIZATION OF PRODUCTION-INVENTORY SYSTEMS	68
C. L. Hendricks, A. J. Koivo, and S. J. Citron, Purdue University	
A SUB-OPTIMAL HEATING SYSTEM	78
Adel H. Eltimsahy, The University of Toledo and Louis F. Kazda, The University of Michigan	
ANALYSIS OF INTERCEPT CONTROL TECHNIQUES WITH COMPUTER GENERATED MOVIES	88
G. S. Willhoff, C. W. Swonger, H. P. Semmelhack and R. C. Friend, Cornell Aeronautical Laboratory	
A CLASS OF DISCRETE SIGNAL DESIGN PROBLEMS IN BURST NOISE . . .	97
Howard I. Silver, Fairleigh Dickinson University and Ludwik Kurz, New York University	
TIME-VARYING AND NONLINEAR SYSTEMS	107
Chairman: N. Wax, University of Illinois	
A THEOREM CONCERNING THE DC EQUATIONS OF NONLINEAR NETWORKS . .	107
I. W. Sandberg and A. N. Willson, Jr., Bell Telephone Laboratories, Inc.	
BOUNDS FOR THE SOLUTIONS OF DC EQUATIONS OF NONLINEAR NETWORKS.	110
I. W. Sandberg and A. N. Willson, Jr., Bell Telephone Laboratories, Inc.	
CHARACTERISTICS OF A SUBOPTIMUM PHASE-LOCK DEMODULATOR	112
John J. Uhren, Jr., University of Notre Dame	
ESSENTIAL NONLINEARITY OF PHASE-SENSITIVE DETECTOR CHARACTERISTICS	122
Branko Leskovar, Lawrence Radiation Laboratory	
CHARACTERIZATION OF TIME-VARYING NONLINEAR SYSTEMS	131
M. Arozullah, Queen's University at Kingston, Canada	
ESTIMATION OF THE SETTLING TIME OF A CLASS OF TIME-VARYING FEEDBACK CONTROL SYSTEMS	133
N. Ahmed, University of Ottawa, Canada and V. Gourishankar, University of Alberta, Canada	
EXACT AND APPROXIMATE SOLUTIONS FOR LINEAR TIME-VARYING SYSTEMS	142
Hooshang Hemami, The Ohio State University	
ON INTEGRAL TRANSFORMS FOR LINEAR TIME VARYING SYSTEMS	151
S. V. Sankaran and S. A. Musa, University of Pennsylvania	
CODING	153
Chairman: F. Preparata, University of Illinois	
TRUNK AND TREE SEARCHING PROPERTIES OF THE FANO SEQUENTIAL DECODING ALGORITHM	153
James L. Massey and Michael K. Sain, University of Notre Dame	

WEIGHT ENUMERATION THEOREMS	161
E. R. Berlekamp, Bell Telephone Laboratories, Inc.	
A SIMPLE METHOD OF DETECTING DECODING ERRORS WITH THRESHOLD DECODING	171
Djimitri Wiggert, University of Massachusetts	
SECURE ELECTRONIC CRYPTOGRAPHY	181
Philip R. Geffe, Westinghouse Electric Corporation	
CAPACITY OF AN ADDITIVE GAUSSIAN COLORED NOISE CHANNEL WITH NOISELESS FEEDBACK	189
Thomas W. Eddy, Bell Telephone Laboratories, Inc. and Jack K. Wolf, Polytechnic Institute of Brooklyn	
SYSTEM THEORY	198
Chairman: C. A. Desoer, University of California, Berkeley	
ON THE CONSISTENCY CONDITIONS AND THE CANONICAL FORM OF FIXED LINEAR DYNAMIC SYSTEMS	198
Anthony J. Grammaticos, New York University	
ON THE IDENTIFICATION OF LINEAR TIME-INVARIANT SYSTEMS FROM INPUT-OUTPUT DATA	207
B. Gopinath, Bell Telephone Laboratories, Inc.	
A CONTINUOUS STOCHASTIC APPROXIMATION ALGORITHM FOR NON-LINEAR SYSTEM IDENTIFICATION	217
D. W. Ricker and G. N. Saridis, Purdue University	
ROUND-OFF ERROR OF FLOATING-POINT DIGITAL FILTERS	219
T. Kaneko and B. Liu, Princeton University	
THE IDENTIFICATION OF LINEAR SAMPLED-DATA SYSTEMS BY DIRECT SEARCH MINIMIZATION	228
G. Leininger and H. R. Martens, State University of New York at Buffalo	
REPRESENTATION OF GENERAL LINEAR NETWORKS	230
R. Saeks, University of Notre Dame	
DERIVATION OF CANONICAL STATE EQUATIONS FOR A CLASS OF DISTRIBUTED SYSTEMS	232
J. A. Resh and I. C. Goknar, Michigan State University	
NETWORK THEORY	237
Chairman: D. A. Calahan, University of Michigan	
TOPOLOGICAL FORMULAS FOR STATE-VARIABLE ANALYSIS OF RLC NETWORKS	237
G. O. Martens, University of Manitoba, Canada	
THE SYNTHESIS OF A CLASS OF TRANSFORMERLESS n-PORT NETWORKS . .	247
Dean J. Schmidlin and Frank J. Lupo, New York University	

A GEOMETRIC INTERPRETATION OF THE PARAMOUNT CONDITION OF RESISTIVE N-PORTS	254
D. J. H. Moore, University of Newcastle, Australia and G. E. Sharpe, The City College of New York and Columbia University	
CLASS INVARIANT GENERATION OF POSITIVE REAL FUNCTIONS	256
N. K. Bose, University of Pittsburgh	
SOME SYNTHESIS METHODS FOR ADJUSTABLE NETWORKS USING MULTI- VARIABLE TECHNIQUES	262
James F. Delansky, The Pennsylvania State University	
ON THE DESIGN OF DOUBLE PASS-BAND FILTERS	264
M. Sablatash, The University of Toronto, Canada	
A LOW LEVEL WIDE RANGE AC LOGARITHMIC AMPLIFIER	274
Robert S. Cuikay, Air Force Cambridge Research Labs., R. E. Sprague and J. Fay, Northeastern University	
GENERAL SESSION B	276
Chairman: T. N. Trick, University of Illinois	
TRANSIENT ANALYSIS OF LOSSY TRANSMISSION LINES	276
Franklin H. Branin, Jr., IBM Systems Development Lab.	
DISTRIBUTED RC NETWORK SYNTHESIS - WHAT IMPORTANT CHALLENGES REMAIN?	286
Ralph W. Wyndrum, Jr., Bell Telephone Laboratories, Inc.	
THE APPLICATION OF THE METHOD OF STEEPEST ASCENT TO A VERY HIGH ORDER OPTIMIZATION PROBLEM: THE PROBLEM OF DEVELOPING EVASION TACTICS FOR AIRCRAFT PURSUED BY MISSILES.	296
B. S. Morgan, Jr., J. F. Schaefer, D. H. deDoe, M. D. Ciletti, and G. Cook, Frank J. Seiler Research Laboratory	
ON ENERGETIC STABILITY OF FEEDBACK SYSTEMS	305
Vaclav Dolezal, State University of New York at Stony Brook and Mathematical Institute of the Czechoslovak Academy of Sciences	
ALGORITHMS FOR COMPUTER-AIDED DESIGN AND ANALYSIS	311
Chairman: J. B. Cruz, Jr., University of Illinois	
STABLE INTEGRATION METHODS FOR ELECTRONIC CIRCUIT ANALYSIS WITH WIDELY SEPARATED TIME CONSTANTS	311
R. H. Allen and C. Pottle, Cornell University	
EFFICIENT NUMERICAL ANALYSIS OF NON-LINEAR CIRCUITS	321
D. A. Calahan, The University of Michigan	
A TWO-SETS-OF-EIGENVALUES APPROACH TO THE COMPUTER ANALYSIS OF LINEAR SYSTEMS	332
I. W. Sandberg and H. C. So, Bell Telephone Laboratories, Inc.	

A METHOD FOR COMPUTING THE TRANSFER-FUNCTION PARTIAL-FRACTION EXPANSION FROM THE STATE EQUATIONS	362
H. C. So, Bell Telephone Laboratories, Inc.	
A GREATEST COMMON DIVISOR ALGORITHM FOR A VECTOR SPACE	374
Thomas G. Marshall, Jr., Rutgers - The State University	
GENERATION OF EQUIVALENT DC ELECTRONIC CIRCUITS BY INTERPOLATION, EXTRAPOLATION, AND INTRAPOLATION	386
E. B. Kozemchak, Bell Telephone Laboratories, Inc., and M. A. Murray-Lasso, Case Western Reserve University	
CONSIDERATIONS FOR OPTIMUM GCA PROGRAM APPLICATION	396
Donnamaie E. Meyerhoff, Northrop Corporate Laboratories, and Lawrence P. McNamee, University of California, Los Angeles	
"INITIAL CONDITIONS" ALGORITHM FOR STATE VARIABLE ANALYSIS OF LINEAR SYSTEMS	406
F. H. Herman, U. S. Army Picatinny Arsenal, Dover, New Jersey and S. D. Bedrosian, University of Pennsylvania	
STOCHASTIC CONTROL	408
Chairman: W. S. Levine, Mass. Institute of Technology	
MINIMAL VARIANCE FEEDBACK CONTROLLERS: INITIAL STUDIES OF SOLUTIONS AND PROPERTIES	408
Stanley R. Liberty and Michael K. Sain, University of Notre Dame	
A NEW ALGORITHM FOR SUBOPTIMAL STOCHASTIC CONTROL	418
Renwick E. Curry, Cornell University	
CONTROL AND ESTIMATION OF LINEAR SYSTEMS WITH NON-GAUSSIAN A PRIORI DISTRIBUTIONS	426
A. V. Cameron, Monash University, Australia	
NONLINEAR PROGRAMMING AND RECURSIVE OPTIMAL ESTIMATION	432
D. Isaacs and W. D. Lochrie, University of California, Irvine	
A STOCHASTIC MODEL-REFERENCE ADAPTIVE CONTROL SYSTEM	442
S. H. Wu and J. E. Brandeberry, Marquette University	
AN APPROXIMATE METHOD OF OPTIMAL CONTROL OF NONLINEAR DYNAMICAL SYSTEMS WITH STATE-DEPENDENT NOISE	444
Yoshifumi Sunahara, Kyoto Institute of Technology, Japan	
STOCHASTIC CONTROL SYSTEMS WITH STATE-DEPENDENT NOISE	454
J. Y. S. Luh and M. Maguiraga, Purdue University	

OPTIMAL CONTROL OF DISCRETE AND DISTRIBUTIVE SYSTEMS	456
Chairman: J. Medanic, University of Illinois and M. Pupin Institute, Belgrade, Yugoslavia	
CONTINUOUS DELAY SYSTEMS UNDER DISCRETE CONTROL	456
Sheldon T. Becker, Bell Telephone Laboratories, Inc. and Philip V. Lopresti, Northwestern University	
DISCRETE SYSTEMS WITH DELAYS IN CONTROL	465
Dong Hak Chyung, University of Iowa	
OPTIMIZATION OF DISCRETE-TIME SYSTEMS BY SUCCESSIVE APPROXIMATION	468
M. H. Richardson and R. J. Leake, University of Notre Dame	
A MAXIMUM PRINCIPLE FOR DISCRETE SYSTEMS WITH CONTROL VARIABLE INEQUALITY CONSTRAINTS	475
Juergen Bruckner and Sherman Wu, Marquette University	
IMPLEMENTING THE SOLUTION TO CONTINUOUS OPTIMAL CONTROL OF DISTRIBUTED PARAMETER SYSTEMS BY FINITE DIFFERENCES . . .	485
George L. Kusic, University of Pittsburgh	
SUFFICIENT CONDITIONS FOR OPTIMAL CONTROL OF A CLASS OF DISTRIBUTED SYSTEMS	496
Hiralal C. Khatri, The Catholic University of America	
THE OPTIMIZATION OF MULTIVARIABLE SYSTEMS WITH DISTRIBUTED PARAMETERS	505
V. J. Tarasov, Western Electric Company, Inc. and H. J. Perlis, Newark College of Engineering	
DIGITAL SYSTEMS	515
Chairman: G. Metze, University of Illinois	
PREEMPTIVE SCHEDULING OF COMPUTATION GRAPHS	515
R. R. Muntz and E. G. Coffman, Princeton University	
BOUNDS ON MEMORY REQUIREMENTS OF MULTIPROCESSING SYSTEMS . . .	523
Y-C. E. Chen and D. L. Epley, University of Iowa	
CONTROLLABILITY OF NONLINEAR SEQUENTIAL NETWORKS	532
Frederic J. Mowle, Purdue University	
MODULE COMPLEXITY AND NAND NETWORK DESIGN ALGORITHMS	538
Edward S. Davidson, Stanford University and Gernot Metze, University of Illinois	
STATE DIAGRAMS OF NONSINGULAR TRANSFORMATIONS ON J_m -MODULES . .	549
H. Louis Althaus, University of Notre Dame	
SYNTHESIS OF STOCHASTIC SEQUENTIAL MACHINES	559
Larry C. Schooley, University of Arizona and Gerry L. Kelly, University of Kansas	

A MARKOV MODEL FOR RELIABILITY OF PROBABILISTIC SEQUENTIAL CIRCUIT WITH BERNOULLI INPUTS	568
G. S. Glinski and Who-Kee Chung, University of Ottawa, Canada	
SENSITIVITY	578
Chairman: B. S. Morgan, Jr., F. J. Seiler Research Lab.	
SENSITIVITY CONSIDERATION OF MULTIPLE INPUT COMPENSATOR DESIGN FOR DYNAMIC OPTIMIZATION	578
Vidojko V. Ciric, Rice University and M. Pupin Research Institute, Belgrade, Yugoslavia, and J. V. Leeds, Rice University	
SENSITIVITY ANALYSIS AND IMPLICIT FUNCTIONS	589
J. M. Holtzman, Bell Telephone Laboratories, Inc.	
GENERATION OF SENSITIVITY FUNCTIONS FOR LINEAR SYSTEMS USING LOW ORDER MODELS	598
Dennis F. Wilkie, Ford Motor Company, and William R. Perkins, University of Illinois	
PERFORMANCE SENSITIVITY AS A TENSOR	615
Lewis Meier, Stanford Research Institute	
MODAL ANALYSIS AND SENSITIVITY REDUCTION IN DISTRIBUTED PARAMETER SYSTEMS	617
C. K. Sanathanan and D. C. Reddy, University of Illinois, Chicago Circle	
LOW-SENSITIVITY DESIGN OF THE LINEAR MINIMAX OUTPUT REGULATOR PROBLEM	619
Ronald D. Bonnell, University of Kentucky and Roger P. Webb and Steven L. Dickerson, Georgia Institute of Technology	
QUANTIZATION, MULTIPARAMETER SENSITIVITY AND ROUND-OFF ERRORS IN DIGITAL FILTERS	621
Thomas G. Mancuso and Samuel C. Lee, New York University	
SENSITIVITY OF STABILITY BEHAVIOR OF LARGE NONLINEAR TIME- VARYING SYSTEMS	632
Ruey-Wen Liu and R. J. Leake, University of Notre Dame	
GENERAL SESSION C	634
Chairman: J. B. Cruz, Jr., University of Illinois	
ON SYSTEMS WITH UNKNOWN OR UNCERTAIN PARAMETERS	634
John B. Thomas, Princeton University	
OPTIMAL CONTROL	646
Chairman: W. R. Perkins, University of Illinois	
ADAPTATION IN A QUANTIZED PARAMETER SPACE	646
Allen Gersho, Bell Telephone Laboratories, Inc.	

THE COMPOSITION PROBLEM FOR DIFFERENTIAL GAMES	654
Michael D. Ciletti, Frank J. Seiler Research Laboratory	
ON THE DESIGN OF OPTIMAL LINEAR SYSTEMS USING ONLY OUTPUT- VARIABLE FEEDBACK	661
William S. Levine and Michael Athans, Massachusetts Institute of Technology	
OPTIMAL LINEAR CONTROL SYSTEMS WITH INCOMPLETE STATE MEASUREMENTS	671
J. D. Ferguson and Z. V. Rekasius, Northwestern University	
THE HUMAN OPERATOR AS AN OPTIMAL CONTROLLER	680
Gary L. Blank, Litton Systems, Inc., and Peter J. Schumacher, Jr., Marquette University	
SYSTEM EQUIVALENT OPTIMAL CONTROL PROBLEMS	690
W. J. Rugh and G. J. Murphy, Northwestern University	
A VECTOR COST FUNCTION FOR ADAPTIVE SYSTEMS	700
N. K. Sinha and V. Temple, McMaster University, Canada	
AN ELIMINATION ALGORITHM FOR COMPUTATION OF THE MINIMAX . . .	702
Juraj Medanic, University of Illinois and M. Pupin Institute, Belgrade, Yugoslavia	
SWITCHING THEORY	711
Chairman: W. J. Poppelbaum, University of Illinois	
A REEXAMINATION OF TERNARY SWITCHING THEORY	711
Anthony S. Wojcik and Gernot Metze, University of Illinois	
TWO-WAY SHIFT-REGISTER REALIZATIONS OF SEQUENTIAL MACHINES . .	722
S. S. Yau and G. F. DePalma, Northwestern University	
AN ALGORITHM FOR DETERMINING THE EQUIVALENCE OF REGULAR EXPRESSIONS	732
John P. Hayes, University of Illinois	
COSET PARTITIONS AND LINEAR SEQUENTIAL CIRCUITS	743
K. C. Wang, Marquette University	
ON THE INDISTINGUISHABILITY OF FAULTS IN DIGITAL SYSTEMS . . .	752
Donald R. Schertz and Gernot A. Metze, University of Illinois	
SYNTHESIS REQUIREMENTS FOR FAULT DETECTION	761
Theo Jay Powell, University of Illinois	
COMMUNICATIONS	773
Chairman: A. H. Haddad, University of Illinois	
ROBUST DETECTION PROCEDURES FOR SIGNALS OF KNOWN FORM IN NEARLY GAUSSIAN NOISE	773
R. D. Martin and S. C. Schwartz, Princeton University	

AN ANALYSIS OF AN AUTOMATIC THRESHOLD UNIT	783
Thomas E. Eger, HRB-Singer, Inc.	
SIMULTANEOUS DETECTION AND ESTIMATION OF PATTERN CHARACTERISTICS IN SELF-LEARNING SYSTEMS	792
Y. T. Chien, University of Connecticut	
THE HIGHER PROBABILITY NEIGHBORHOOD DECISION RULE	800
Gordon Chang, S. J. Dwyer, III, and John J. Komo, University of Missouri, Columbia	
VARIABLE-DATA-RATE PROCEDURES FOR FEEDBACK COMMUNICATION SYSTEMS	802
John J. Metzner and Kenneth C. Morgan, New York University	
INTERPOLATION OF A GAUSSIAN MARKOV PROCESS	812
Michael A. Schumer, Raytheon Research Division	
GRAPH THEORY	814
Chairman: W. Mayeda, University of Texas at Austin (on leave from the University of Illinois)	
SOME IMPROVED ALGORITHMS FOR COMPUTER SOLUTION OF THE TRAVELING SALESMAN PROBLEM	814
K. Steiglitz and P. Weiner, Princeton University	
TOPOLOGICAL FORMULATION OF NETWORK FUNCTIONS WITHOUT GENERATION OF K-TREES	822
W. R. Dunn, Jr. and S. P. Chan, University of Santa Clara	
SELF-DUAL GRAPHS AND DIGRAPHS	832
Narsingh Deo, Jet Propulsion Laboratory	
DIAKOPTICS - KRON'S METHOD OF TEARING	841
K. U. Wang, University of Cincinnati	
ALGEBRAIC GENERATION OF NETWORK FUNCTION COEFFICIENTS AND PARTIAL DERIVATIVES	843
Gerry L. Kelly, University of Kansas	
STABILITY OF NONLINEAR SYSTEMS	845
Chairman: I. W. Sandberg, Bell Telephone Laboratories, Inc.	
THE EFFECT OF STRAY PARAMETERS ON THE STABILITY OF A CLASS OF NON-LINEAR NETWORKS	845
S. T. Sathé and T. E. Stern, Columbia University	
L^P -STABILITY ($1 \leq p \leq \infty$) OF NONLINEAR TIME-VARYING FEEDBACK SYSTEMS	852
M. Y. Wu and C. A. Desoer, University of California, Berkeley	
PRACTICAL STABILITY OF DYNAMICAL SYSTEMS	860
A. N. Michel, Iowa State University	

COMPUTER AIDED RC DISTRIBUTED CIRCUIT DESIGN	862
K. W. Heizer, Southern Methodist University and Collins Radio	
TRANSIENT RESPONSE OF A CLASS OF ACTIVE LUMPED-DISTRIBUTED SYSTEMS	870
J. J. Bourquin and J. S. Tzeng, University of Missouri at Rolla	
ANALYSIS OF TWO BRANCHES OF THE ROOT LOCUS WHICH ORIGINATE FROM A SIMPLE POLE IN A SYSTEM CONTAINING DISTRIBUTED LAG	880
B. J. Ball, Mississippi State University	
A SYNTHESIS ALGORITHM FOR CASCADED TRANSMISSION LINES	889
A. Gersho, Bell Telephone Laboratories, Inc. and B. Kinariwala, University of Hawaii	

ALGORITHMS FOR PATTERN CLASSIFICATION*

R. L. KASHYAP
Purdue University
Lafayette, Indiana

CONTENTS

- I. General nature of pattern classification problems
 - 1.1: An approach to the abstraction problem
- II. "Deterministic" algorithms with correctly classified samples
 - 2.1: Minimum error scheme
 - 2.2: Modified least squares scheme
 - 2.3: Other algorithms and comparison
- III. Algorithms with samples of unreliable classification
- IV. Stochastic algorithms
- V. Discussion

I. GENERAL NATURE OF PATTERN CLASSIFICATION PROBLEMS

A number of problems in systems theory can be formulated in terms of pattern classification. For example, the methods of recognition of aircraft from radar signals, the development of optical character recognition devices which can decipher handwriting, and the methods of geophysical prospecting based on seismic recordings are intimately connected with pattern classification. The aim of pattern classification can be roughly stated as the determination of the class to which a certain sample belongs, given that the sample belongs to one of the finite number of classes under consideration. But a general statement of the problem involves the consideration of three fundamental aspects namely characterization, abstraction and generalization. These shall be considered successively.

Even though the classes are clearly differentiated by specific qualities, the choice of the independent variables which describe the classes is not obvious. Therefore, we must first decide on a procedure for determining the relevant independent variables known as the primary attributes or features of the problem. The process of extracting the features is referred to as the characterization. Many a time the characterization is performed in two steps. In the first step, usually referred to as feature extraction, all the relevant candidates for features are listed. For example, a two-dimensional visual pattern can be characterized in terms of $n \times n$ grid of optical cells and the interstices of the n^2 cells are candidates for the features. Usually this operation is straight forward. The second step involves the selection of the features (including their number) among the various candidates using a suitable index of performance. This is largely an unsolved problem. It is needless to say that the features chosen should be amenable for automatic processing and computation. The selected features will be labeled as $x_1, x_2, \dots, x_m$.

After the features have been selected, we have to determine a decision rule for classifying a new sample based on its pattern vector x . We can choose the decision rule by minimizing a suitable performance criterion. In many problems, the available information can be nothing more than a finite set of samples from every class, the so-called training samples. The process of obtaining the decision rule is called as abstraction. The basic point here is the abstraction of the information contained in the training samples into a compact functional form.

Once the decision rule has been determined, it will be used to

*This work was partially supported by NSF under Grant GK-1970 and NASA under contract No. NGR 15-005-021.

categorize the new sample patterns (of unknown classification). The ability of the decision rule or classifier to determine correctly the class of the new samples is called the generalization property of the device. The generalization property is best stated in probabilistic terms like the probability of correct classification.

The three aspects of the classification problem are not completely independent of each other. Clearly the choice of improper attributes manifests itself in an unduly complex form for the decision rule to obtain a certain level of performance. Similarly, the ability to generalize may be considered as the criterion for getting the decision function during the abstraction of the problem. The three aspects essentially serve as guidelines in getting the desired solution to the over-all problem. In this paper the main attention will be on the abstraction aspect of the problem, with brief comments on the generalization aspect.

1.1 An Approach to the Abstraction Problem

We will restate the abstraction problem. We will consider here only the situations in which there are only 2 classes ω_0 and ω_1 . With every pattern vector x with known classification, we will associate a variable y whose value uniquely indicates the observed classification of the sample. For example, y is +1 or -1 if the associated pattern x belongs to ω_1 or ω_0 respectively. The only available information about the classes is the sample pairs $\{x(i), y(i)\}$, $i=1, 2, 3, \dots$ which can be assumed to be drawn from a population with probability density $p(x, y)$. Based only on this information, we should arrive at a decision rule (1.1) with decision function $d(x)$ which partitions the m -dimensional feature space into 2 regions; i.e.,

$$\begin{aligned} \text{if } d(x) > \epsilon > 0, x \text{ is classified in } \omega_1 \\ < \epsilon, x \text{ is classified in } \omega_0 \end{aligned} \quad (1.1)$$

Let us first indicate the form of the optimal decision rule if the joint density $p(x, y)$ is known. The decision function which minimizes the probability of error (assuming the same penalty for all types of errors) is given below [1]:

$$d(x) = \{p(y=1|x)/p(y=-1|x) - 1\} \quad (1.2)$$

The conditionals $p(y=1|x)$ and $p(y=-1|x)$ are to be calculated from the conditional probability densities of the samples of each class viz. $p(x|y=1)$ and $p(x|y=-1)$ and the apriori probabilities of the classes $p(y=+1)$ and $p(y=-1)$. Since $p(x, y)$ is not known, we usually in the first instance assume that they are multivariate normal distributions with unknown means and covariances. These unknown parameters are estimated from the given samples. In eq. (1.2) we can substitute the estimated probability functions for the exact probability functions and obtain a decision rule (1.1).

Let us briefly consider some of the major objections to this point of view. Firstly, in many problems, especially those in which the pattern vector can assume only a finite number of values, the assumption of normality is invalid. Even in many problems where the pattern vectors assume continuous values like the classification of crops, the conditional density plots may not even be bell shaped. Secondly, the number of parameters to be estimated, i.e., the means and the covariances, is $m^2 + 3m$ in the 2-class problem where m is the number of features. Thus, the results obtained from this method are of very little value unless the number of available sample pairs $\{x(i), y(i)\}$ is very much greater than $m^2 + 3m$. Thirdly, the prior probabilities $p(y)$ can rarely be estimated with any accuracy unless the available number of samples is very large. Needless to say, the decision scheme does not minimize the probability of misclassification except perhaps in an asymptotic sense.

All these objections point to the need for alternate methods of posing the classification problem. In this paper the decision functions will be chosen by minimizing suitable criterion functions involving only the decision rule and the available samples. We will develop recursive techniques for the determination of the decision function and discuss their properties when the number of available samples is finite or infinite. It is important to note that these techniques do not need the explicit knowledge of the probability density of patterns $p(x)$ or the apriori class probabilities $p(y)$.

It is convenient to restrict the decision functions $d(x)$ to linear combinations of certain known linear independent functions $\varphi_1(x), \varphi_2(x) \dots$ in the variables $x_1, \dots, x_m$.

$$d(x) = \sum_{i=1}^n \alpha_i \varphi_i(x) \quad (1.3)$$

where $\alpha_1, \dots, \alpha_n$ are unspecified coefficients. The principle reason for this assumption is that it allows the analysis and computation to be carried out in the finite dimensional space of the vector α instead of the infinite dimensional space of the decision functions. The second reason is that the functions $\varphi_1(x), \varphi_2(x), \dots$ can be chosen to reflect the special characteristics of the classes under consideration more clearly than the features $x_1, \dots, x_m$ themselves whose choice depends heavily on the data processing considerations. By choosing the number n sufficiently large, we can get as close to the true optimal decision function as desired. The functions $\varphi_1(x), \varphi_2(x), \dots$ are usually chosen among the members of the various families of orthogonal polynomials like Legendre, Hermite, etc. By making n go to infinity in eq. (1.3), we can get the true, optimal decision function.

We have already mentioned that the classification problem is completely described by the conditional probability function $p(y/x)$. Depending on its nature we can divide the classification problems into two categories:

Case (i) : Deterministic classification: $p(y/x)$ is 0 or 1

Case (ii): Probabilistic classification: $0 < p(y/x) < 1$.

In the deterministic problems, the classes are non-overlapping and the randomness arises solely due to x . For example, this is a reasonable assumption while dealing with the recognition of numbers 0-9 with properly chosen features. In the probabilistic classification problems, any pattern may belong to all the classes simultaneously with probabilities $p_0, p_1, \dots$ respectively, which sum to one. A typical example is the case of detection of stochastic signal in the presence of noise. We will discuss the 2 cases separately. In the former case, we will be dealing with separating functions and their approximations whereas in the latter case, we will deal with the function $E(y|x)$ and its approximations. Since both cases deal with the approximation of functions, the algorithms have certain similarity in structure. At the end, we will comment on their relative merits.

A brief word on notation is in order. φ by itself means $\varphi(x)$. $\varphi(i)$ stands for $\varphi(x(i))$. Matrix $A > 0$ means matrix A is positive definite. $\lambda_{\min}(A)$ means minimum eigenvalue of A , similarly $\lambda_{\max}(A)$. Vector $\beta > 0$ means $\beta_i > 0 \forall i$. $\|A'\|$ means spectral norm A . Prime denotes a transpose.

II. "DETERMINISTIC" ALGORITHMS WITH CORRECTLY CLASSIFIED SAMPLES

We will deal here only with problems involving 2 non-overlapping classes w_0 and w_1 with a separating function $f(x)$ obeying eq. (2.1).

$$\begin{aligned} f(x) &> \epsilon > 0 & \forall x \in w_1 \\ < -\epsilon < 0 & \forall x \in w_0 \end{aligned} \quad (2.1)$$

Clearly the separating function $f(x)$ is not necessarily unique. The only available information is the sample pairs $\{(x(i), y(i)), i=1, 2, \dots\}$ where $y(i)$ is the true classification variable associated with the pattern $x(i)$

When the number of distinct samples $\{x(i), y(i)\}$ is finite and when x assumes continuous values, there is no question of "recovering" the separating function from the samples even when $f(x)$ has a finite linear expansion. However, when x assumes a finite number of values, it may be possible to recover the separating function if the classification of all the members of the 2 classes are given. Since, in general, nothing is known about $f(x)$, we will aim at obtaining a "best" approximation to the separating function. For computational ease we will restrict the approximating functions to forms given in eq. (2.2).

$$d(x) = \sum_{i=1}^n \alpha_i \varphi_i(x) = \alpha' \varphi(x), \quad n < \infty \quad (2.2)$$

where the parameters $\alpha_1, \alpha_2, \dots$ are to be determined. With every candidate $d(x)$, we can associate a performance index $J(\alpha)$ to indicate the deviation of $d(x)$ from a separating function $f(x)$. The function $J(\alpha)$ can be evaluated with the aid of the available sample pairs $\{x(i), y(i), i=1, 2, \dots\}$. Our intention is to find α^* , the optimal value of α by minimizing $J(\alpha)$ with respect to α and thus find the best approximating function $f(x) = \alpha^* \varphi(x)$ based only on the available sample pairs.

We will give here some useful performance indices and the corresponding $J_1(\alpha)$ minimization algorithms.

2.1 Minimum Error Scheme [2], [3]

The performance index $J_1(\alpha)$ is the expected value of the number of samples that are misclassified by the decision rule (2.4) involving the function $d(x)$.

$$\begin{aligned} d(x) > 0 &: \text{place } x \text{ in class } \omega_1 \\ < 0 &: \text{place } x \text{ in class } \omega_0 \end{aligned} \quad (2.4)$$

$$J_1(\alpha) \triangleq E[V(\alpha, x)] = E[(1 - \text{sgn}(\alpha' \varphi(x)y))] \quad (2.5)$$

We will write down a first order gradient method to minimize $J_1(\alpha)$.

$$\alpha(i+1) = \alpha(i) - \rho(i) \nabla_{\alpha} J_1(\alpha) |_{\alpha = \alpha(i)} \quad (2.6)$$

Since $\nabla_{\alpha} J_1$, the gradient of $J_1(\alpha)$ with respect to α , cannot be evaluated, we replace $(-\nabla_{\alpha} J_1(\alpha))$ by the measurable quantity $(-\nabla_{\alpha} V(\alpha, x))$ where

$$-\nabla_{\alpha} V(\alpha, x) = \text{negative gradient of } J_1(\alpha) = (1 - \text{sgn}(\alpha' \varphi(x)y)) \varphi(x)y \quad (2.7)$$

Hence, the algorithm (2.6) becomes

$$\alpha(i+1) = \alpha(i) + \rho(i)(1 - \text{sgn}(\alpha' \varphi(i)y(i))) \varphi(i)y(i) \quad (A-1)$$

and

$$\sum_{i=1}^{\infty} \rho(i) = \infty, \quad \sum_{i=1}^{\infty} \rho^2(i) < \infty \quad (2.8)$$

The gain sequence $\rho(i)$ in eq. (2.8) is chosen as shown above to "damp" the error caused by using the gradient of $V(\alpha, x)$ instead of the gradient of $E(V(\alpha, x))$ in (A-1). If the samples $\varphi(i)$, $i=1, 2, \dots$ are independent and have finite mean square value, then the algorithm (A-1) converges to the value α^* which is the solution of eq. (2.9).

$$E[(\text{sgn } f(x) - \text{sgn}(\alpha' \varphi(x))) \varphi(x)] = 0 \quad (2.9)$$

In view of eq. (2.7), α^* can also be regarded as minimum of the criterion function $J_1(\alpha)$.

If the function $f(x)$ has the following finite linear expansion $f(x) = \sum_{i=1}^n \theta_i \varphi_i(x)$, then the best approximation $\alpha^* \varphi(x)$ will be a separating function; i.e., $\text{sgn}(\alpha^* \varphi(x)) = \text{sgn } f(x)$. Moreover, in this case, the gain $\rho(i)$ in (A-1) can be a constant.

All the results given above assume the availability of infinite number

of samples. When we have only a finite number N of distinct samples, we can either stop the algorithm (A-1) after N iterations or continue further by cycling the available samples using the formula $\varphi(j) = \varphi(i)$ if $j \equiv i \pmod{N}$. In either case, the algorithm leads to only an approximate minimum of $J_1(\alpha)$. In particular, if the given N samples are separable in the n -dimensional φ -space, i.e., the inequalities (2.10) are consistent, the algorithm (A-1) with a constant gain $\rho(i)$ and cycling of the available samples leads to a solution α^* of the inequalities (2.10) in a finite number of steps. The scheme described above is the original perceptron algorithm [2].

$$\alpha' \varphi(i) y(i) > 0, \quad i=1, \dots, N \quad (2.10)$$

2.2 Modified Least Squares Scheme [4]

The criterion function $J_2(\alpha, \beta)$ in this scheme involves only the available samples which should be finite in number. Thus the approximation obtained here is optimal whether the number of samples is large or small, in contrast to the earlier scheme.

In this scheme, we will assign positive weights $\beta_1, \beta_2, \dots, \beta_N$ to the N samples $\varphi(1), \dots, \varphi(N)$ and vary them, in addition to varying α , to obtain optimality. Recall that $\alpha' \varphi(i)$ is the projection of the vector $\varphi(i)$ on the hyperplane $\alpha' \varphi(x) = 0$. We desire a hyperplane such that the projections of the members of the same class should be close to each other and those of different classes should be away from each other; i.e., we desire all the weighted projections $\alpha' \varphi(i) / \beta_i$ of the samples belonging to class w_1 to have a value around $+1$ and those of class w_0 to have a value around -1 . The performance index measures the deviations of the projections from their assigned values ± 1 .

$$\begin{aligned} J_2(\alpha, \beta) &= \sum_{x(i) \in w_1} \beta_i^2 \left(1 - \frac{\alpha' \varphi(i)}{\beta_i}\right)^2 + \sum_{x(i) \in w_0} \beta_i^2 \left(-1 - \frac{\alpha' \varphi(i)}{\beta_i}\right)^2 \\ &= \|A\alpha - \beta\|^2 \end{aligned} \quad (2.11)$$

where

$$A = \left[\begin{array}{c} \varphi'(1)y(1) \\ \vdots \\ \varphi'(N)y(N) \end{array} \right] \quad \beta = \left[\begin{array}{c} \beta_1 \\ \vdots \\ \beta_N \end{array} \right]$$

The minimizing algorithm is given below:

$$\left. \begin{aligned} \alpha(i+1) &= \alpha(i) + \rho S A' |\epsilon(i)|, \quad \epsilon(i) = A\alpha(i) - \beta(i) \\ \beta(i+1) &= \beta(i) + \epsilon(i) + |\epsilon(i)|, \quad \beta(0) > 0 \end{aligned} \right\} \quad (A-2)$$

The scalar ρ , and matrix S satisfy the condition (2.12).

$$\rho = \text{constant} > 0, \quad S = S', \quad (\rho^2 S A' A S - 2\rho S) > 0 \quad (2.12)$$

The sequence $\alpha(i)$, $\beta(i)$ given by (A-2) converge respectively to the values α^* , β^* (not necessarily unique) which minimize $J_2(\alpha, \beta)$. If, in addition, the inequalities $A\alpha > 0$ are consistent, the convergence occurs in a finite number of steps and α^* will be a solution of the inequalities $A\alpha > 0$. Some of the typical choices of S and ρ giving rise to varying degrees of speed and complexity are given below.

- (i) $S = (A'A)^{-1}$, $0 < \rho < 2$, $\alpha(0)$ arbitrary
- (ii) $S = I$, $0 < \rho < (\|A'A\|^{-1})$, $\alpha(0)$ arbitrary
- (iii) $S = (2\|A'A\|I - A'A) / \|A'A\|^2$, $0 < \rho < 2$, $\alpha(0)$ arbitrary

In case (i), if $\alpha(0) = (A'A)^{-1} A' \beta(0)$, and $\beta(0) = (1, 1, \dots, 1)$, then $\alpha(i)$ represents the best linear least squares fit for given $\beta(i)$, i.e., minimizes

$J_2(\alpha, \beta(i))$. Hence $\alpha(i) = (A'A)^{-1}A'\beta(i)$.

2.3 Other Algorithms and Comparison

In addition to the algorithms we have already given, we have to mention the relaxation algorithm of Agmon [5], the Wong-Eisenberg algorithm [6] and the method of linear programming [7]. It is also possible to develop iteration schemes in function space to recover $f(x)$ directly without additional assumptions. Let us discuss the various algorithms. When the inequalities $A\alpha > 0$ are consistent, then all the algorithms achieve a solution of the inequalities. However, case (i) of the algorithm (A-2) seems to possess the highest convergence rate among them, excluding linear programming. The apparent reason is the introduction of the quantities $\beta(i)$ as variables in the algorithm (A-2). Heuristically, the variable vector β allows those row constraints (in the inequalities) which are not satisfied to have different weights at every iteration. Comparison of the algorithm (A-2) with the method of linear programming is inconclusive, in general. However, in most cases, $A\alpha > 0$ will be inconsistent and in such cases the linear programming method is of little use. In such circumstances, the choice lies between the algorithms (A-1) and (A-2), depending on the number of the available samples and computational complexity. If the number of the samples is small, it may be advisable to use (A-2) the least squares schemes (A-2) since the criterion function $J_1(\alpha)$ involving an expectation is not exactly relevant here. On the other hand, if the number of samples is large, the minimum error scheme (A-1) may be preferable to (A-2) since the latter is more complex than the former and it (algorithm (A-2)) involves the storage of all the samples.

III. ALGORITHMS WITH SAMPLES OF UNRELIABLE CLASSIFICATION

In this case, we assume as before that there is a separating surface $f(x) = 0$ separating the 2 classes. But we assume that the classifications of the pattern vectors $x(i)$ given to us are erroneous. The available sample pairs are only $\{x(i), z(i)\}$, $i=1, 2, \dots$ where $z(i)$ is the observed (noisy) classification of the pattern. In other words, $z(i) = (\text{sgn } f(x(i)))\eta(i)$ where $\eta(i)$ is a sequence of independent noise variables assuming one of 2 values, ± 1 . This seems to serve as a valid model in many problems like alpha numeric recognition.

To derive an optimal approximation for the unknown function $f(x)$, let us use the criterion function $J_1(\alpha)$ of section (A) mentioned earlier. The corresponding algorithm (A-3) is given below where $\hat{\eta} = E(\eta)$ is known and $\hat{\eta} \neq 0$.

$$\alpha(i+1) = \alpha(i) + \rho(i)[z(i) - \hat{\eta} \text{sgn}(\alpha'(i)\varphi(i))]\varphi(i)/\hat{\eta} \quad (\text{A-3})$$

with $\sum \rho(i) = \infty$, $\sum \rho^2(i) < \infty$.

The surprising thing is that (A-3) converges to the same value α^* as algorithm (A-1) which uses only correctly classified samples. Thus, as long as we know $\hat{\eta}$, the mean of the measurement noise η , and if $\hat{\eta} \neq 0$, then we can recover the minimum error approximation $\alpha^{*'}\varphi(x)$ to $f(x)$. The only difference is that algorithm (A-3) may be slower than (A-1). When $\hat{\eta} = 0$, the samples have practically no information about their classification, and hence it is easy to guess that the algorithm will not work in this case.

IV. STOCHASTIC ALGORITHMS

In this section we will consider the two class problem under the condition that the classes are not mutually exclusive. Any pattern x can belong to class w_1 with probability $p(y=1|x)$ and to class w_0 with probability $p(y=-1|x) \triangleq 1-p(y=1|x)$ where $0 < p(y=1|x) < 1$. Using the given

sample pairs $\{(x(i), y(i)), i=1, 2, \dots\}$ with unknown probability density $p(x, y)$, we would like to construct a decision function $g(x)$ to classify any new sample; i.e.,

$$\begin{aligned} \text{if } g(x) > 0 & \quad x \text{ is placed in class } \omega_1 \\ < 0 & \quad x \text{ is placed in class } \omega_0 \end{aligned}$$

A natural choice for $g(x)$ is that function which minimizes the expected risk. Let us call this decision function $g^0(x)$ and express it in terms of the conditional probability density $p(y|x)$.

$$g^0(x) = E(y|x) - c \triangleq (p(y=1|x) - p(y=-1|x)) - c$$

where the threshold c is a known function of the costs associated with the 2 types of errors. We will obtain a best approximation to the function $E(y|x) \triangleq f(x)$ directly on the basis of the available samples without making any explicit assumptions on its structure.

As before, the approximations to $f(x)$ are constrained to be of the form in eq. (4.1).

$$d(x) = \sum_{i=1}^n \alpha_i \varphi_i(x) = \alpha' \varphi(x) \quad (4.1)$$

We would like to determine the α^* , the optimal value of the coefficient vector α by minimizing a suitable criterion function and study the relation between $f(x)$ and the optimal approximation $\hat{f}(x) = \alpha^{*'} \varphi(x)$.

We will show later that if $f(x)$ indeed has a finite expansion of the form in eq. (4.1), the best approximation $\hat{f}(x)$ will equal the actual function $E(y|x) = f(x)$. This situation arises, for example, when the conditional probability densities $p(x|y=1)$ and $p(x|y=-1)$ are multivariate normal. Then the function $f(x)$ will be a quadratic function in the variables $x_1, \dots, x_m$. Hence, if we chose the functions $\varphi_i(x)$ among all the linear and quadratic terms among $1, x_1, \dots, x_m$, then we can recover the optimal discriminant function by the method of this section. Thus, the methods of this section are at least as good as the classical theory when the conditional density $p(x|y)$ is known and appear to be better than the others when the conditional density $p(x|y)$ is unknown.

Since we are approximating the function $E(y|x)$, any criterion function used here must involve the error $(y - \alpha' \varphi(x))$. We will consider here only the mean square error criterion

$$J_3(\alpha) = E(y - \alpha' \varphi(x))^2 \quad (4.2)$$

It is easy to show that

$$\alpha^* = \underset{\alpha}{\text{Arg}}[\text{Min } J_3(\alpha)] = [E(\varphi(x)\varphi'(x))]^{-1} E[\varphi(x)f(x)] \quad (4.3)$$

Further, if $f(x) = \sum_{i=1}^n \theta_i \varphi_i(x)$, then $\alpha^* = \theta$.

To recover the α^* from the available samples, we will use the first order gradient algorithm [9]. Recall that

$$-\nabla_{\alpha} J_3(\alpha) = \varphi(x)(y - \alpha' \varphi(x))$$

Hence, the stochastic descent scheme is described in (A-4) and the associated gain sequence $\rho(i)$ in eq. (4.5).

$$\alpha(i+1) = \alpha(i) + \rho(i)\varphi(i)(y(i) - \alpha'(i)\varphi(i)) \quad (A-4)$$

$$\sum_{i=1}^{\infty} \rho(i) = \infty, \quad \sum_{i=1}^{\infty} \rho^2(i) < \infty \quad (4.5)$$

If the samples $\{x(i), i=1, 2, \dots\}$ are independent with finite mean square value and $E(\eta^2(i)) < \infty$, then $\alpha(i)$ tends to α^* with probability one and in mean square. If one desires an algorithm with a better rate of convergence, we will construct a stochastic second order gradient algorithm [9] given in (A-5). It involves the use of the symmetric transformation matrix $\Sigma(i)$ on

the gradient $\nabla_{\alpha} J_3(\alpha)$.

$$\alpha(i+1) = \alpha(i) + \Sigma(i+1)\varphi(i)(y(i) - \alpha'(i)\varphi(i))$$

$$\Sigma(i+1) = \Sigma(i) - \frac{\Sigma(i)\varphi(i)\varphi'(i)\Sigma(i)}{1 + \varphi'(i)\Sigma(i)\varphi(i)} \quad (A-5)$$

Under the same conditions as before, $\alpha(i)$ will converge to α^* and $\Sigma(i)$ to $[E(\varphi\varphi')]^{-1}$. Algorithm (A-5) normally converges to a neighborhood of α^* faster than (A-4). For a discussion of their convergence rates see [10] and [11].

In many instances the assumption of the independence of the samples $x(i)$ is hard to justify. If we know that $f(x)$ has a finite linear expansion, then the independence assumption can be replaced by the following weaker condition $\sum_i \|x(i)\|^2 < \infty$ and another which can be stated heuristically as follows: the samples $x(1), x(2), \dots$ must repeatedly span all the directions of the Euclidean m -space. For details see [9] and [12].

We finally remark that it is possible to recover $f(x)$ directly by iterating in the function space.

V. DISCUSSION

We have presented a series of algorithms (A-1)-(A-5) for getting an optimal decision function based only on the given sample pairs. The principle question in practice is the choice of the algorithm for a given classification problem. Clearly the characterization of the classification problem as "deterministic" or "probabilistic" is subjective unless the number of samples is infinite. A possible clue to the choice of the algorithm is given by the available number of samples.

If the available number of samples is small, it is hard to justify the use of criterion functions involving expectations. Hence, the only algorithm suitable is (A-2), case (I) since it corresponds to a least square fit. Note that the result given by (A-5) is the same as that given by the first step of (A-2), case (I). When the number of samples is large, we can use any of the algorithms (A-1), (A-4) and (A-5). Note all of them are error correcting algorithms. Both (A-1) and (A-4) are strongly similar in that the term $\alpha'(i)\varphi(i)$ bounded by +1 and -1 in (A-4) corresponds to the term $\text{sgn}(\alpha'(i)\varphi(i))$ in (A-1). It is conceivable that the results given by these algorithms (A-1), (A-4) and (A-5) may be close to one another.

The concept introduced in section III involving classification problems with deterministic separating surfaces and misclassified samples seems very realistic in many problems like photograph processing

However, the corresponding algorithm (A-3) will be of use only when some method of estimating the bias η can be found.

REFERENCES

This paper is a preliminary version of Chapter 3 in the book entitled "Adaptive and Learning Systems" by K. S. Fu and J. M. Mendel, to be published by Academic Press, 1969. (Ed. 1969)

1. C. W. Helstrom, Statistical Theory of Signal Detection, Pergamon Press, 1960.
2. A. Novikoff, On Convergence Proofs for Perceptions, Proceedings of Symposium on Mathematical Theory of Automata, Vol XII, Polytechnic Institute of Brooklyn, 1963, pp. 615-622.

3. E. M. Braverman, "On the Potential Function Method," Automation and Remote Control, Vol. 26, 1965, pp. 2205-2213.
4. Y. C. Ho and R. L. Kashyap, "A Class of Iterative Procedures for Linear Inequalities," J. SIAM Control, Vol. 4, No. 1, 1966.
5. S. Agmon, The Relaxation Method for Linear Inequalities, Canad. J. Math., 6(1956), pp. 382-392.
6. E. Wong and E. Eisenberg, Iterative Synthesis of Threshold Functions, J. Math. Anal. Appl., Vol. 11, 1965, pp. 226-235.
7. G. Dantzig, "Linear Programming and Extensions", Princeton University Press, 1963.
8. R. L. Kashyap, "Pattern Classification and Switching Report", Ph.D. Thesis, (1965), Harvard University and Tech. Rept., 483, Cruft Laboratory.
9. C. C. Blaydon and Y. C. Ho, "On the Abstraction Problem in Pattern Classification," Proc. National Electronic Conf., 1966, pp. 857-862.
10. R. L. Kashyap and C. C. Blaydon, "Recovery of Functions from Noisy Measurements Taken at Randomly Selected Points and Its Application to Pattern Classification", Proc. IEEE, Vol. 54, No. 8, Aug. 1966, pp. 1127-1128.
11. T. J. Wagner, "Rate of Convergence of an Algorithm for Recovering Functions from Noisy Measurements Taken at Randomly Selected Points", IEEE Trans. on System Sciences and Cybernetics, July, 1968.
12. A. A. Albert and L. A. Gardner, Jr., Stochastic Approximation and Nonlinear Regression, MIT Press, 1967.

K. S. FU, P. J. MIN^{*} and T. J. LI
Purdue University
Lafayette, Indiana

ABSTRACT

The problem of feature selection in pattern recognition is briefly reviewed. Feature selection techniques discussed include (i) information theoretic approach, (ii) using direct estimation of error probability, (iii) feature-space transformation, and (iv) approach of using stochastic automata model. These techniques are applied to the selection of features in crop classification problems. Computer simulation results are presented and compared.

I. INTRODUCTION

One important problem in pattern recognition is the selection of effective features from a given set of feature measurements. It is known that the performance of a pattern recognition system is closely related to the feature measurements taken by the classifier. On the other hand, the use of a large number of feature measurements will definitely increase the complexity or the size of the classifier. In practice, the maximum number of features is usually constrained, though the number of possible feature measurements is large. The selection of a smaller size feature sub-set from a given set of features becomes a problem very often encountered by the system designer. The effectiveness of features should relate to the performance of the recognition system, usually in terms of the probability of correct recognition. However, in some cases, if more a priori information is known (e.g., the feature distributions for each pattern class), an indirect measure of feature effectiveness (e.g., divergence, entropy function, etc.) may be used for feature selection. In another approach, the feature effectiveness is measured in terms of the feature's capability to characterize the patterns to be recognized. In the following, several important feature selection techniques are briefly presented and the results from their application to the crop classification problem compared.

II. INFORMATION THEORETIC APPROACH

Approaches from the viewpoint of information theory have been suggested for evaluating the feature effectiveness. Both the divergence [1] and the average information [2,3] about the pattern classes characterized by the features have been proposed as feature selection criteria. The application of divergence as a criterion for feature selection is summarized in this section. Assume that for pattern class ω_i , $i=1, \dots, m$, the feature vector $X = [x_1, \dots, x_N]'$ is distributed according to a multivariate Gaussian density function with mean vector M_i and covariance matrix K ; i.e., $p(X/\omega_i) \sim N(M_i, K)$. Let $L = \log[p(X/\omega_i)/p(X/\omega_j)]$, then

$$L = X'K^{-1}(M_i - M_j) - \frac{1}{2} (M_i - M_j)'K^{-1}(M_i - M_j) \quad (1)$$

$$\text{and} \quad E[L/\omega_i] = \frac{1}{2} (M_i - M_j)'K^{-1}(M_i - M_j) \quad (2)$$

The divergence between ω_i and ω_j is defined as [4]

$$J(\omega_i, \omega_j) = E[L/\omega_i] - E[L/\omega_j] = (M_i - M_j)'K^{-1}(M_i - M_j) \quad (3)$$

[†]This work was supported by the National Science Foundation Grant GK-1970 and USDA Contract No. 12-14-100-8926(20).

^{*}P. J. Min is presently with IBM, Kingston, New York.

It is noted that in (3) if $K=I$, the identity matrix, then $J(\omega_i, \omega_j)$ represents the squared distance between M_i and M_j .

If a Bayes' decision rule is used for the classifier, then for $P(\omega_i) = P(\omega_j) = 1/2$, X is from the class ω_i , i.e., $X \sim \omega_i$ if $L > 0$, or $X \sim \omega_j$ if $L < 0$. Then the probability of misclassification is

$$e_{ij} = \frac{1}{2} P \{L > 0 / \omega_j\} + \frac{1}{2} P \{L < 0 / \omega_i\} \quad (4)$$

From (1), (2) and (3), it is concluded that $p(L/\omega_i)$ is a Gaussian density function with mean $(1/2)J$ and variance J . Similarly, $p(L/\omega_j)$ is also a Gaussian density function with mean $(-1/2)J$ and variance J . Using (4), it can be easily shown that [1]

$$e_{ij} = \int_{(1/2)\sqrt{J}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{1}{2} y^2 \right] dy \quad (5)$$

which means that e is a monotonically decreasing function of $J(\omega_i, \omega_j)$. Therefore, features selected according to the magnitude of $J(\omega_i, \omega_j)$ will imply their corresponding discriminatory power between ω_i and ω_j .

For more than two pattern classes, the criterion of maximizing the minimum divergence or the expected divergence between any pair of classes has been proposed for feature selection [5,6]. A feature selection criterion based on a general separability measure has recently been suggested for multiclass pattern classification problems [8]. Basically, consider that, in a multiclass classification problem, the performance of classification can be expressed in terms of a weighted sum of all pairwise probability of misclassifications; i.e.,

$$e = \frac{2}{m(m-1)} \sum_{i=1}^m \sum_{j=1}^m e_{ij} \quad (i \neq j) \quad (6)$$

Suppose that the feature vectors for each pattern class are Gaussian distributed with mean vectors M_i and covariance matrices K_i , $i=1, \dots, m$, and a family of linear discriminant function $g_i(X) = B_i'X - c_i$, $i=1, \dots, m$, is used for classification. Let

$$g_i(X) - g_j(X) = B_{ij}'X - c_{ij} \quad (7)$$

Then $e_{ij} = (1/2)P \{B_{ij}'X > c_{ij} / \omega_j\} + (1/2)P \{B_{ij}'X < c_{ij} / \omega_i\}$

$$= (1/2)[2 - P\{\xi < d_j\} - P\{\xi < d_i\}] \quad (8)$$

where ξ is the standardized Gaussian random variable, and

$$d_i = \frac{B_{ij}'M_i - c_{ij}}{(B_{ij}'K_i B_{ij})^{1/2}}, \quad d_j = \frac{c_{ij} - B_{ij}'M_j}{(B_{ij}'K_j B_{ij})^{1/2}} \quad (9)$$

Since, in this case, for linear classification procedures, the minimax procedure is admissible [10], it is used here to determine an appropriate criterion for feature selection in multiclass classification problems. From (6) and (8), the condition of minimizing the maximum of e is $d_i = d_j = d_{ij}$, and then

$$d_{ij} = \frac{B_{ij}'(M_i - M_j)}{(B_{ij}'K_i B_{ij})^{1/2} + (B_{ij}'K_j B_{ij})^{1/2}} \quad (10)$$

Equation (8) becomes

$$e_{ij}^* = [1 - P\{\xi < d_{ij}\}] \quad (11)$$

It is noted from (11) that a monotonic functional relationship exists between e_{ij}^* and d_{ij} . The quantity d_{ij} is denoted as the separability measure between ω_i and ω_j . Based on the separability measure d_{ij} , the feature selection criterion proposed is to maximize the expected separability

measure for all pairs of classes.

Since the divergence reflects only the mean information for discriminating classes, it is a suboptimum measure of feature effectiveness in the case of unequal covariance matrices, and it is difficult to find an explicit function relating the divergence to the probability of pairwise misclassification. Using minimax linear discriminants, however, it is possible to find a monotonic functional relation between the probability of misclassification and the separability measure. Also, since only the separability measure between a pair of classes is considered, the feature selection technique based on linear discriminants is the best approximation for that based on the maximum likelihood decision rule by means of the optimality of the minimax condition.

The proposed feature selection technique in multiclass pattern recognition has been applied to crop classification problems. The data used were multispectral measurements taken from airborne scanning radiometers [11]. The measurements (features) consist of 12 electrical signals, of which each one is proportional to the radiant energy from the scene in a different wavelength band. The 12 bands cover the range from 0.4 to 1.0 microns in the visible and near infrared portions of the spectrum. By simultaneously sampling the output of 12 bands, one obtains a vector which contains all the spectral information available about a given resolution element on the ground. From histogram studies, the data can be fairly reasonably modeled by multivariate normal distributions with unequal covariance matrices. The Bayes' decision rule with equal a priori probabilities is used as the classifier. The statistical parameters were estimated from 425 samples per class.

After using the proposed technique to select the best feature sets from all possible combinations, the effectiveness of the sets was tested by computing the probability of error (misclassification) with 7,530 samples (about 1,500 samples per class) by the classifier. The results of this experiment shown in Figure 1 indicate that the optimum feature sets were selected for all the combinations. A typical example of the recognition results is shown in Table 1. The following points are observed from the experimental results: (1) It is possible that a smaller number of subsets could be almost as effective as the complete feature set. This is probably due mainly to the deviation of actual feature distribution from the Gaussian assumption and the error involved in the estimation of statistical parameters. Estes [12] has shown similar results for equal covariance matrices. (2) The experiments were extended to include the selection of the best feature sets by using the divergence as the selection criterion. The results are less effective than the separability measure of the minimax linear discriminant, but the time required for computation is shorter and the accuracy is acceptable for most cases.

III. DIRECT ESTIMATION OF ERROR PROBABILITY

In many pattern recognition problems, the knowledge about feature distributions from which the measure of feature effectiveness can be developed is usually not available. Even if the probability structure of feature measurements can be approximated by simple known distributions (e.g., normal distributions with unequal covariances), it is difficult to find the exact analytical measure of feature effectiveness which directly reflects the error probability involved. In these cases, nonparametric methods for feature selection are needed. In the following, a nonparametric feature selection technique based on the direct estimation of error probability is briefly described.

The proposed feature selection criterion is based on the direct estimation of the minimum expected error from a given data set of training samples and is independent from the classification technique used. The problem of estimating probability density function has received much attention in the statistical literatures. The direct application of these density estimation

techniques is often impractical for discriminatory analysis because of large memory storage requirements. However, since we are only considering the estimation of the minimum error, the techniques for estimating density function can be used in a nonparametric method for feature selection.

Based on the results of Parzen and Cacoullos [15,16], the estimation of the density function at activate point Z in multivariate p -dimensional discrete measurement space can be derived

$$\hat{p}_n(Z) = \frac{1}{n} \sum_{i=1}^n U \left(\frac{[(Z-X_i)(Z-X_i)']^{1/2}}{\sigma(n)} \right) \quad (12)$$

where X = the sample feature vector, n = number of training samples, 2σ = interval of quantized discrete level of measurement, $U(y) = 1$ if $y < 1$, or 0 if $y \geq 1$.

For most engineering problems, however, the parameter $\sigma(n)$ is a given constant and there is no way to control it. In other words, for a given σ , we can only increase the number of training samples n so that the estimates of (12) is asymptotically unbiased. Therefore, for relatively large sample problems,

$$c(Z) = \sum_{i=1}^n U \left(\frac{[(Z-X_i)(Z-X_i)']^{1/2}}{\sigma} \right) \quad (13)$$

is called a cell of the histogram and then the conditional probability at the activate point Z for a given class ω_j becomes

$$P(Z/\omega_j) = \frac{c_j(Z)}{n_j} \quad (14)$$

Bayes' decision rule is used for classification, and the feature selection criterion is to be developed on the basis of the direct estimation of the minimum misclassification. Hence, the Bayes rule directs the choice of the most probable classes for a given feature vector. The resultant probability of over-all error estimated by n training samples is

$$\bar{P}_n(\epsilon) = \sum_{i=1}^s \left\{ \sum_{j=1}^m P(Z_i/\omega_j) P_j - \max_j [P(Z_i/\omega_j) P_j] \right\} \quad (15)$$

where $P(Z_i/\omega_j)$ = the conditional probability for a given class ω_j estimated by (14) with n_j training samples from class ω_j , P_j = a priori probability of class ω_j , m = number of pattern classes, and s = the total number of discrete points in the p -dimensional feature space.

A direct estimation of (15) is proposed as a feature selection criterion [8]. That is, the feature set α_k is considered more effective than the feature set α_ℓ if

$$\bar{P}_n(\epsilon/\alpha_k) < \bar{P}_n(\epsilon/\alpha_\ell) \quad (16)$$

Some engineering limitations arise in implementing the proposed procedure.

(A) In the calculation of (15), a large amount of computation time is usually required especially when the number of classes is large. However, considering a pair of classes each time, considerable computation time can be saved and the feature selection procedure suggested in [8] is directly applicable for multiclass pattern recognition problems using pairwise probability of misrecognition. (B) As in the case of some parametric feature selection procedures, the major problem in selecting the optimum feature set is that there are too many feature sets to compare and hence the computation time required is sometimes rather long. In this case, the sequential forward selection procedure and sequential backward selection procedure can be used as suboptimal procedures for feature selection [8,11].

The proposed nonparametric method of feature selection technique was

applied for the crop classification problem which was also tested for the parametric feature selection criterion in the experiment of section II. 423 samples from each class are used for the estimation of density functions. The results of this experiment are shown in Table 2. Up to 2-feature sets, the best feature set is selected. However, for 3-feature sets, the feature sets $\{x_1, x_6, x_{12}\}$ and $\{x_1, x_9, x_{11}\}$ are equally effective. In this case, the approximation procedure for small sample problems can be applied to distinguish them [8]. The results show also that all the classes are separable for most 4-feature sets (41 sets).

The effectiveness of the selected best feature subsets are compared with that of the parametric feature selection technique and tested by computing the percentage of misclassification using 7,530 test samples (about 1,500 samples per class).

IV. FEATURE-SPACE TRANSFORMATION

Several authors have studied the linear feature-space transformation techniques to apply for the feature selection problems. Watanabe [17] introduced the feature-space compression technique based on Karhunen-Loeve (K-L) expansion. Fu and Chien [18,19] generalized the transformation technique for extracting the effective features and carried out the application of the generalized K-L expansion for feature selection problems to character recognition problems. The algorithm resulting from this generalized expansion is summarized here.

Let $X^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \dots, x_N^{(i)}\}$ be a random variable from class w_i with zero mean. The K-L expansion allows the expression of a random process in terms of a set of orthonormal vectors. Any component of the sample vector $X^{(i)}$ can be expressed as a function of the orthonormal set of vectors $\{Y_k; k=1, 2, \dots, N\}$

$$x_\ell^{(i)} = \sum_{k=1}^N \gamma_k^{(i)} y_{k\ell} \quad (17)$$

where $y_{k\ell}$ is the ℓ th component of the k th orthonormal vector Y_k and $\gamma_k^{(i)}$ is a coefficient of $y_{k\ell}$ for class w_i .

If the generalized expansion in (17) exists for all $x_\ell^{(i)}$; $\ell=1, 2, \dots, N$ and the coefficients satisfy the conditions:

$$\begin{aligned} \sum_{i=1}^m p_i E \left[\gamma_k^{(i)} (\gamma_\ell^{(i)})' \right] &= \sigma_k^2 & \text{if } k = \ell \\ \sum_{i=1}^m p_i E \left[\gamma_k^{(i)} (\gamma_\ell^{(i)})' \right] &= 0 & \text{if } k \neq \ell \end{aligned} \quad (18)$$

then the component of the ensemble covariance function $K = \|c_{k\ell}\|$ becomes

$$c_{k\ell} = \sum_{i=1}^m p_i E \left[x_k^{(i)} x_\ell^{(i)} \right] = \sum_{k=1}^N \sigma_k^2 y_k y_k' \quad (19)$$

The expression in (17), whose coordinates $\{Y_k; k=1, 2, \dots, N\}$ are determined by (19) through the covariance function $K = \|c_{k\ell}\|$, will be called the generalized Karhunen-Loeve expansion. Here $\{\sigma_k^2; k=1, 2, \dots, N\}$ are the eigenvalues of the covariance function K and $\{Y_k; k=1, 2, \dots, N\}$ are eigenvectors.

It can be shown that the orthonormal coordinate system produced by the K-L expansion minimizes the entropy function and that the discriminant information over the ensemble of the probability space is concentrated in few coordinates by a linear transformation. Referring to [18], the transformation procedure is basically to construct a new coordinate system in which the coordinate coefficients are mutually uncorrelated over the ensemble of all pattern classes. In many recognition problems where covariance functions are real and symmetric, this decorrelation process simply amounts to the diagonalization of the covariance function. Similar to the truncated

K-L expansion, the method of Hotelling's principal components also utilizes linear transformation to obtain the uncorrelated random variables [20].

The feature selection technique based on generalized K-L expansion was tested on the crop classification problem. The results were compared with those using parametric feature selection technique discussed in section II. The data used in these experiments are the same as that for feature selection techniques in the previous sections. Bayes' decision rule is used as the classifier, and appropriate statistical parameters are estimated from training samples for each class.

The effectiveness of transformed feature spaces was tested by again computing the percentage of misclassification using 7,530 test samples. In Table 3, the results are compared with the best feature subspaces selected by the parametric feature selection techniques in the experiment of section II. The transformed p-dimensional feature space is less effective than the same dimensional feature subspace for all $p(< N)$, but the difference in performance for the 4-feature space is only 1.3%. The computation time required is much shorter for the transformation technique.

V. STOCHASTIC AUTOMATA APPROACH

In the theory of automata games, a class of learning automata with intelligent behavior in random environments, are proposed [21]. The random environments are specified either by a set of statistical parameters or by some game rules. It is noted that the learning automata can be used as a feature selection scheme if the automata are defined in terms of feature subsets and the environments are characterized by training samples and a certain decision rule. Here we present the structure of the stochastic learning automata first and then consider the application of this learning model as a feature selection scheme.

Consider the interaction between an automaton and a random environment C which is characterized by $C(p_1, p_2, \dots, p_k)$ where $0 \leq p_i \leq 1$ for $i=1, \dots, k$. The input y of the automaton can take two values: $y=1$ (penalty) and $y=0$ (non-penalty). The output f of the automaton is its action and q is the internal state. In one experiment, if the automaton takes action f_i , $i=1, 2, \dots, k$, then the next input y becomes

$$y = \begin{cases} 1 & \text{with probability } p_i \\ 0 & \text{with probability } 1-p_i \end{cases}$$

It is required to design an automaton so that the expected penalty is minimized if the p_i 's are unknown but fixed constants.

A stochastic learning automaton $B_{r,k,b}$ has been proposed [21] to achieve this purpose. This $B_{r,k,b}$ model with k actions corresponding to k states, has memory capacity r , randomization factor b , $0 \leq b < 1$, and a state transition rule as follows: (Figure 2)

1. change the state from q_i to q_{i+1} for $i=1, 2, \dots, (k-1)$, from q_k to q_1 if $y(n)=1$ and $q(n) \neq q(n-1)$
2. with probability $(1-b)$ change state in the same fashion if $y(n)=y(n-1)=\dots=y(n-r+1)=1$ and $q(n)=q(n-1)=\dots=q(n-r+1)$.

Otherwise, stay in the same state. $y(n)$ and $q(n)$ indicate the input and state of the automaton at the n th experiment respectively.

It has been shown [21] that the expected penalty for model $B_{r,k,b}$ operating in the random environment $C = C(p_1, p_2, \dots, p_k)$ is

$$M(B_{r,k,b}, C) = \frac{\sum_{i=1}^k (1-p_i \cdot b) \frac{1}{p_i^{r-1}}}{\sum_{j=1}^k (1-p_j \cdot b) \frac{1}{p_j^r}} \quad (20)$$

It is noted that this function is monotonically decreasing with respect to both memory capacity r and randomization factor b . The "expedient

behavior" is preserved even with $r=1$, $b=0$. That is, the strategy due to model $B_{r,k,b}$ is always better than a purely random strategy, or

$$M(B_{r,k,b}, C) < \sum_{i=1}^k p_i / k$$

with $r=1, 2, \dots$, $0 \leq b < 1$, $0 < p_i \leq 1$, $i=1, 2, \dots, k$. Furthermore, asymptotic optimality is achieved in the sense that

$$\lim_{r \rightarrow \infty} M(B_{r,k,b}, C) = \text{Min}(p_1, p_2, \dots, p_k).$$

In other words, during the experiment the probability of applying the best action approaches 1 as the memory capacity r increases indefinitely.

Due to the learning property of the $B_{r,k,b}$ model, this model can be employed as a feature selection scheme. Each possible subset of the feature measurements is considered to be an action taken by the automaton. The input of the automaton is 1 or 0 depending upon whether an incorrect classification or a correct classification is made. If the state transition rule of $B_{r,k,b}$ model is used, then the optimal action, or the feature subset corresponding to the minimum probability of misrecognition, will be selected most frequently. Relative effectiveness of the feature subsets can be measured by their relative frequencies of occurrence in the experiment.

As a practical example, the multispectral data obtained for crop classification problems are used to demonstrate this feature selection scheme. For simplicity, consider the problem of finding the best 3-feature combination from a set of six feature measurements $X_1, X_6, X_9, X_{10}, X_{11}, X_{12}$. There are $C_3^6 = 20$ possible subsets. Let $f_{i,j,k}$ with $i, j, k = 1, 2, \dots, 12$, $i \neq j$, $j \neq k$ denote the action of choosing the subset consisting of feature measurements X_i, X_j, X_k .

From the data available, a decision between soybean and corn is made for every input sample. The learning automaton $B_{r,k,b}$ receives a penalty for an incorrect classification and a non-penalty for a correct classification. Memory capacity r is chosen to be 5 with $b=0$. 2000 samples are used for classifications and state transitions. The computational result indicated that the most frequently used action is $f_{6,9,11}$ with the relative frequency 0.52. This implies that the three feature combination (X_6, X_9, X_{11}) is the best combination in the sense that the probability of misclassification is minimized. Other combinations of the feature measurements can be tested by increasing the size of the action set.

VI. COMPARISONS AND FURTHER REMARKS

Using minimax linear discriminant functions, a separability measure was derived for the problem of feature selection in parametric multiclass pattern recognition when features from each class are distributed according to Gaussian probability density function with unequal covariance matrices. In general, when covariance matrices are unequal and the over-all misclassification is used as a measure of feature effectiveness, it is believed that the separability measure based on minimax linear discriminants could be a useful criterion of feature selection in multiclass pattern recognition. The proposed feature selection technique has been successfully applied to crop classification problems.

The proposed nonparametric feature selection technique based on direct error estimation has been successfully applied to the crop classification problem. Since no assumptions are made for the probability structure of feature distributions and the independence of measurements, the proposed nonparametric technique is more effective than the parametric feature selection technique when there is no prior knowledge of feature distributions for each class. Moreover, when the number of training samples is small, the nonparametric method based on the density approximation gave comparatively the best experimental results. The computation time required for the nonparametric method of feature selection techniques is, in general,

more than that for the parametric method.

The feature selection technique based on generalized K-L expansion has shown good results for the crop classification problem. This technique is superior due to the fact that the transformation procedure is less dependent on the probability structure of classes which, in practice, can only be estimated by a limited number of training samples. However, the optimality of the technique is defined over the ensemble of classes, and it makes no explicit provision for discriminatory analysis between classes. For all $p \leq m$, the transformed p -dimensional feature space was less effective than the same dimensional feature subspace selected by the parametric feature selection technique based on linear discriminant functions and the nonparametric feature selection technique based on direct error estimations. When class distributions are unknown, the feature-space transformation technique is superior to the parametric feature selection technique in multiclass pattern recognition. Considering computation time required, the feature-space transformation technique has some advantages over the proposed nonparametric feature selection technique based on direct error estimations. However, to obtain an over-all optimum performance of pattern recognition system, the nonparametric method of feature selection technique is preferred.

The main advantage of the stochastic automata model for feature selection is its simplicity in implementation. Furthermore, the approach is also nonparametric in nature and able to match the decision rule used by the classifier.

It might be pointed out that at present there is no unique feature selection technique for all the possible multiclass pattern recognition problems. However, when class distributions can be fairly reasonably approximated by Gaussian distributions, the parametric feature selection technique gives at least some reasonable evaluation of feature effectiveness. And even if the optimum-seeking procedure based on the separability measures of parametric feature selection technique may not provide a unique selection for an optimum feature set in some cases, it is still valid to reduce a large number of feature sets to the smaller number of feature subsets so that the nonparametric feature selection technique based on direct error estimation can be more easily applied to choose the best feature set from this small number of feature subsets. Using the combined method, we may be able to solve many practical feature selection problems in multiclass pattern recognition.

REFERENCES

1. Marill, T. and Green, D. M., "On the Effectiveness of Receptor in Recognition Systems", IEEE Trans. on Information Theory, Vol. IT-9, No. 1, pp. 11-17, January 1963.
2. Lewis, P. M., "The Characteristic Selection Problem in Recognition Systems", IRE Trans. on Information Theory, Vol. IT-8, pp. 171-178, 1962.
3. Barabash, Yu. L., "On Properties of Symbol Recognition", Engineering Cybernetics, No. 5, pp. 71-77, 1965.
4. Kullback, S., Information Theory and Statistics, Wiley, New York, 1959.
5. Grettenberg, T. L., "Signal Selection in Communication and Radar System", IEEE Trans. on Information Theory, Vol. IT-9, pp. 265-275, October, 1963.
6. Fu, K. S. and Chen, C. H., "Sequential Decisions, Pattern Recognition and Machine Learning", TR-EE 65-6, School of Electrical Engineering, Purdue University, April, 1965.
7. Kadota, T. T. and Shepp, L. A., "On the Best Finite Set of Linear Observables for Discriminating Two Gaussian Signals", IEEE Trans. on Information Theory, Vol. IT-13, pp. 278-284, April 1967.

8. Fu, K. S. and Min, P. J., "On Feature Selection in Multiclass Pattern Recognition", TR-EE 68-17, School of Electrical Engineering, Purdue University, 1968.
9. Kailath, T., "The Divergence and Bhattacharyya Distance Measures in Signal Selection", IEEE Trans. on Comm. Tech., Vol. COM-15, No. 1, February 1967.
10. Anderson, T. W. and Bahadur, R. R., "Classification into Two Multivariate Normal Distributions with Different Covariance Matrices", Ann. Math. Stat., Vol. 33, No. 2, pp. 420 ~ 431, June 1962.
11. "Remote Multispectral Sensing in Agriculture", Vol. 1, 2, and 3, The Laboratory for Agricultural Remote Sensing, Purdue Univ., Lafayette, Ind.
12. Estes, S. E., "Measurement Selection for Linear Discriminants Used in Pattern Classification", IBM Research Report RJ-331, San Jose, April 1965.
13. Hughes, G. F., "On the Mean Accuracy of Statistical Pattern Recognizers", IEEE Trans. on Information Theory, Vol. IT-14, No. 1, 1968.
14. Allais, D. C., "The Selection of Measurements for Prediction", Technical Report No. 6103-9, Stanford Electronics Laboratories, November, 1964.
15. Parzen, E., "On Estimation of a Probability Density Function and Mode", Ann. Math. Stat., Vol. 33, pp. 1065 ~ 1076, 1962.
16. Cacoullos, T., "Estimation of Multivariate Density", Univ. of Minnesota, Report No. 40, Dept. of Statistics, May 1964.
17. Watanabe, S., "A Method of Self-Featuring Information Compression in Pattern Recognition", Bionics Symposium, 1966.
18. Fu, K. S. and Chien, Y. T., "On the Finite Stopping Rules and Nonparametric Techniques in a Feature-Ordered Sequential Recognition System", TR-EE 66-16, School of Electrical Engineering, Purdue University, 1966.
19. Chien, Y. T. and Fu, K. S., "Selection and Ordering of Feature Observations in a Pattern Recognition System", Information and Control, Vol. 12, May 1968.
20. Rao, C. R., Advanced Statistical Methods in the Estimation Statistical Parameters, Wiley, New York, 1952.
21. Fu, K. S. and Li, T. J., "On the Behavior of Learning Automata and Its Applications", TR-EE 68-20, School of Electrical Engineering, Purdue University, 1968.
22. Fu, K. S., Sequential Methods in Pattern Recognition and Machine Learning, Chapter 2--Feature Selection and Feature Ordering, Academic Press, New York, 1968.

CLASS	NO OF SAMPS	PCT CORCT	NO. OF SAMPLES CLASSIFIED INTO				
			SOYB	CORN	OATS	WHEAT	CLOV
SOYB	1535	96.0	1473	44	18	0	0
CORN	1476	94.0	77	1397	2	0	0
OATS	1483	90.0	7	4	1334	21	117
WHEAT	1538	97.8	2	0	32	1504	0
CLOV	1498	96.9	5	24	18		1451
TOTAL	7530	95.0	1564	1469	1404	1525	1568

TABLE 1: THE RECOGNITION MATRIX OF EXPERIMENT I
USING 4-TUPLE FEATURE SET $\{x_1, x_6, x_{10}, x_{11}\}$

Table 2. The Results of Nonparametric Feature Selection Technique

Number of Features	Nonparametric Method		Parametric Method	
	Best Feature Set	% Error	Best Feature Set	% Error
1	x_9	33.8	x_9	37.6
2	x_1, x_9	3.1	x_1, x_9	10.6
3	x_1, x_{10}, x_{11}	0.1	x_1, x_{10}, x_{11}	5.0
	x_1, x_9, x_{12}	0.1		
4	41 feature set	0.0	x_1, x_6, x_{10}, x_{11}	4.9

Table 3. Comparison of Feature Selection Method for Transformation Technique

Transformation Technique		p-tuple	Parametric Method	
No. of Dimension	% Recog.		Best Feature Set	% Recog.
1	48.2	1	x_9	62.4
2	75.6	2	x_1, x_9	89.4
3	92.8	3	x_1, x_{10}, x_{11}	95.0
4	93.8	4	x_1, x_6, x_{10}, x_{11}	95.1
5	94.2	5	$x_1, x_6, x_9, x_{10}, x_{11}$	95.6

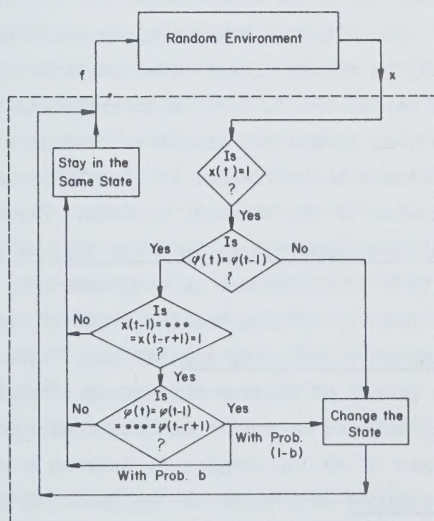
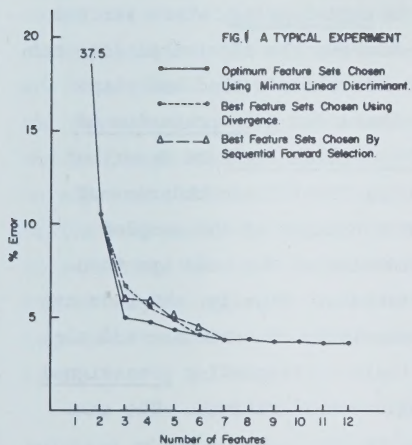


FIGURE 2.

ON THE APPLICATIONS OF PATTERN RECOGNITION TO
GEOPHYSICAL AND GEOLOGICAL PROBLEMS*

Z. J. NIKOLIC
Esso Production Research Company
Houston, Texas

A brief review of certain recently reported applications of pattern recognition techniques in exploration geophysics and geology is presented.

1. Many problems of exploration geophysics can only be solved by an integration of several geophysical methods [1]. In such cases, the interpretation of the collected data is made more efficient and accurate when certain time-consuming tasks are done on a digital computer. Pattern recognition methods can be used in the design of computer programs for joint interpretation of the geophysical data obtained in exploration surveys for oil and minerals.

2. A geophysical survey consists of measurements by which subsurface geological structures or bodies can be detected and located. Very often certain pertinent physical parameters of the subsurface under investigation have to be determined. The information about the subsurface structure is usually needed because of certain known association between the structure and the sought minerals (for instance, some oil fields are found in anticlinal folds and domes [2]).

3. A better geological description of a subsurface structure under study is almost always obtained when a larger number of pertinent geophysical parameters is used in interpretation. In well-logging, where various physical parameters within a borehole are measured, the electrical logs can sometimes be sufficient to identify sandstone in certain sand and clay deposits of the borehole profile. Simple schemes for interpretation of digitized apparent resistivity logs or self-potential logs are described in [3]. The computer is programmed to identify first the boundaries of the beds by finding the intervals of monotonic changes of the sampled apparent resistivity log (AR log). The boundaries of the beds are then the points of maximum variations in such intervals. Finally, the beds are only defined between such points where the magnitude of variation and the values of AR log samples in between exceed their corresponding preassigned thresholds determined on the basis of geologic considerations. The same

*This paper is based mainly on a literature survey and it does not necessarily reflect work at the Esso Production Research Company.

program can interpret the sampled self-potential log (SP log) when two thresholds are properly selected. This program for establishing the boundaries of the beds on the basis of a single curve is only reliable interpretation tool when the beds are homogeneous. A scheme for computer interpretation of the Micro-type resistivity logs (microinverse and micro-normal curves) was described in [4]. This program is designed to recognize (identify) porous beds. Geological and geophysical studies at the exploration site are first used to determine quantitative features of a prototype of the porous beds. The prototype is a point in a three dimensional feature space. The features are: 1) the difference between the caliper and borehole diameter, 2) resistivity read from the microinverse curve, and 3) resistivity read from the micronormal curve. The value of resistance which separates the resistance readings of the microinverse curve into "low" and "high" is preset ("low" values correspond to the porous beds). The program first examines the sampled microinverse curve and selects some representative porous beds. The prototype of porous beds is then the point in the feature space with the coordinates equal to the average of the corresponding coordinates of the representative beds. The remaining unknown beds are classified either into a pattern class of porous beds or a pattern class of unporous beds. Such classification depends on the euclidean distance in feature space between the point which corresponds to the unknown bed and the prototype ("large" distance, where "large" is quantitatively prespecified, would mean that the unknown bed is considered as nonporous). "Self-teaching" is continued by computations of the average coordinates of all the points which were classified into the same pattern class; i.e., two prototypes, one for porous beds and one for unporous beds, are determined. Each newly considered unknown bed is then classified into the pattern class of porous beds if it is closer in the feature space to the prototype of porous beds. Otherwise, the unknown bed is proclaimed to be unporous. It should be mentioned that the quality of obtained interpretation very much depends on the type of initial information which is used in selecting the representative samples of the porous beds. The clustering of samples may also be quite different if some other segments of the logs are first chosen to determine the representative samples of a pattern class.

4. More flexible schemes of pattern recognition are needed in the solution of complicated exploration problems [5,6]. A computer program which can evaluate the information content of different geophysical parameters could point out the most profitable exploration techniques for

certain subsurface geological structures. It was reported [5,7] that a computer program was trained to distinguish the sandstone beds with oil from those sandstone beds with water. The data used in recognition of unknown beds were derived from the well-logs (apparent resistivity, gamma log, neutron-induced gamma, the bed thickness, diameter of the well, the resistance of the borehole fluid, etc.). The program was trained on the data of some known oil-bearing and water-bearing beds. The ordering of geophysical parameters is performed by the program according to their information content (i.e., the relative frequency of observation of the parameter during the recognition). The importance of different geophysical parameters varied as the learning of more difficult to interpret beds was attempted by the program. The resistivity method became less reliable, and the importance of gamma and neutron-induced gamma-logs has increased in identification (recognition) of oil-bearing beds. The performed study with such a learning pattern recognition program has also disclosed certain complex relationships between the measured geophysical parameters.

5. The tasks of the same learning pattern recognition program may be quite different. When classifying the rocks on the basis of their lithology, the pattern classes can be, for example, shale, sandstone, limestone, etc.; when determining different physical properties, they can be, for instance, layers of low, medium, and high porosity. The correlation of borehole profiles is also a problem similar to pattern recognition: layers in different boreholes which belong to the same stratigraphic formation have to be placed in the same pattern class. In such a case the feature vector may be formed from the well-logs which characterize a given segment of the borehole profile. Quantitative determination of physical parameters is also possible. For instance, sandstones of different ranges of porosity can be considered as samples of different pattern classes [6].

It should be realized that such computer programs are only flexible tools for geophysicists and geologists. They permit the users to consider simultaneously a large amount of quantitative information, and test their hypotheses accurately and efficiently. The accuracy of measured parameters, the nature of parameters, and any additional a priori information of geologic and economic nature should be taken into consideration when the results of classification, obtained by the program, are analyzed by the users. In other words, geophysicists and geologists must provide adequate information for finding the feature space in which the classification is going to be performed. Since they are also responsible for evaluation and interpretation of the obtained results, they must be familiar with the

basic theoretical assumptions of the decision rules employed in the process of classification, and with the features formed from the given experimental data.

6. In interpretation of magnetic anomalies it is of interest to find which anomaly corresponds to the ore. With features formed from the measurements of the magnetic fields, and a previous training of the program on anomalies of both types (which correspond to the ore and non-ore geological situations), the computer can perform the classification of an unknown anomaly according to the prespecified decision rule of the program. Several decision rules and many sets of features may have to be assigned to the program, and the obtained results of classifications, not being identical, must be reexamined by the interpreter unless it is a priori known that the employed decision rule corresponds to the statistical descriptions of both pattern classes in the feature space [35].

A search for proper diagnostic information in predicting the quality of oil (according to the percentage of sulfur) was also based on one of possible decision rules and criteria for selection of optimal features [8]. From eleven initial parameters which describe the properties of a reservoir rock and a source rock, the program indicated an essential importance of only seven (the number and nature of the initial set of parameters and those finally proposed by the program as optimal may differ from one oil-field to another). Three pattern classes were considered: 1) the pattern class corresponding to oil with less than 0.5% of sulfur, 2) the pattern class which corresponds to oil with 0.5% - 1% of sulfur, and 3) the pattern class which corresponds to oil with more than 1% of sulfur. The training samples for the program were only used as the test samples in the performed experiments. Low percentage of correct predictions indicates that the selected decision rule was not appropriate for the distributions of given learning samples in the feature space. Perhaps some other method of training and recognition [34,35] could offer a better performance.

7. Digital computers are now used extensively by geologists. The concepts of mathematical statistics, time-series analysis and discrete mathematics are employed in defining the models of geological structures and processes for the studies on digital computers [9-11, 16-18, 21]. The methods of clustering analysis are used in defining the pattern classes which correspond to different geological units [9, 11, 12, 20, 22]. Usually the samples which optimize certain intraset distance are considered as the members of a cluster [23].

A clustering scheme based on the "moving sphere" in the feature space was examined experimentally in [22]. The center of the "moving space" is always set in the "center of gravity" for all points (the points could have different "weights") which were in the interior of the sphere at the previous stage of iteration. When the "moving sphere" becomes motionless, one cluster is obtained and all the points belonging to the interior of the "moving sphere" are removed. The clustering of the other points with a "moving sphere" positioned at a new location is continued. The iteration process ends when all the sample points are removed from the sample space. The distance measure is then used in classification of unknown samples. The sensitivity of clustering on the radius of the "moving sphere" was also investigated experimentally. This clustering scheme was applied to a problem of grouping the elements of a collection of fossils. The scheme is simple, but its success would depend upon whether the obtained clusters can be used to describe unambiguously the pattern classes under consideration (each cluster should contain the samples of one pattern class only; several clusters may correspond to the same pattern class).

8. The comparisons of different geologic surfaces (structural horizons, maps, etc.) are also carried now on digital computers. The features used for quantitative description of the surfaces can be formed from the coefficients of the polynomial trend surfaces [11, 14] or using the eigenvectors of the surfaces [15]. Several measures of similarity are available [11], and the choice will depend on the problem at hand.

The selection of features for description of local uplifts was discussed in [13]. The learning pattern recognition program was trained on the equal number of sites with known local uplifts and the sites which did not possess them.

9. The seismic methods [1,2] provide the most detailed information about the subsurface structures. Elastic waves created at or near the surface of the earth propagate through the rocks and are partially reflected and refracted back to the surface at the interfaces where the acoustical properties of the rock change. The vibrations of the surface at a finite number of locations are recorded and then processed on digital computers. The elements of statistical communication techniques are broadly used in processing the geophysical data [31-33], particularly in the attempts to remove various interferences [28] which almost always mask the primary reflections on the seismic recordings. Such interferences have to be first removed by proper filtering operations before the interpretation in terms

of subsurface geological structures can begin. For instance, a computer program described in [25] can be used for detection of arrival times of seismic signals in the microseismic noise. From the time-distance and amplitude-distance curves one may then solve the inverse problem of finding (identifying) velocity-depth curves which satisfy the observed experimental data [29]. Several solutions exist and their number is reduced if some new data are taken into consideration (for instance the thermal data [30]). The solutions of inverse problems in geophysics [26-29] are usually non-unique. Such is the case of inverse problems with the data obtained by the seismic methods. Resolution of ambiguities in the solutions will be only possible by complex (joint) interpretation of data obtained by different exploration methods. It is not yet clear what to use as the training samples and which pattern classes to define for a learning pattern recognition program in the case of seismic data.

10. The success of pattern recognition methods in solving geological and geophysical problems will depend upon several factors:

a) The geologists and geophysicists must search for adequate, informative parameters in describing geologic structures (systems). Even some seemingly irrelevant parameters may prove important in the solutions of more complicated problems since the selection of features by a learning pattern recognition program is based on the available geological and geophysical information.

b) The storage of such information has to be in the form which can be used with different classification rules. We have to be able to derive certain approximations of the probability density functions in the feature space for all pattern classes of interest.

c) The samples of one pattern class may only be available (or be more numerous in comparison to the number of samples from other pattern classes under consideration) as the training samples of a learning pattern recognition program. The training in such situations can be performed on the basis of clustering techniques [11, 22] but then the choice of the decision rule is made in a heuristic manner by a trial search. It is not yet known which training schemes should be considered in the case when, for instance, only rock samples which do not contain ore of interest are only available. Similar situation exists in the seismic methods where more often a mixture of a signal and numerous interferences than a signal which can be assigned to a particular pattern class is available. Studies of such schemes of

training have just started with certain attempts to formulate the problem [20, 21].

d) Geologists and geophysicists have to become acquainted with the theoretical foundation on which the available decision rules are based, and the limitations of such rules with respect to the available feature space. Experiments with the learning pattern recognition programs will only give indications which rule is applicable and appropriate for the solution of a given problem.

e) The data related to different pattern classes must be collected since the most reliable schemes of training and classification rules are selected when known samples of all pattern classes are used in training. For instance, data related not only to the ore-bearing rocks, but also to those which do not contain ore of interest (at least not in economic quantities) have to be collected by geologists and geophysicists.

With the existent digital computer programs for trend analysis, statistical analysis, time-series analysis, plotting and contouring programs [11], the geophysicists and geologists can perform many time-consuming tasks of data analysis and display more accurately and economically. With the learning pattern recognition computer programs they can also conduct efficiently certain complex interpretations of large quantities of geophysical and geological data. Such interpretations are usually temporary since new information about geological systems under study may indicate that the employed decision rules and the features used for description of the pattern classes were not suitable. The learning pattern recognition programs will then have to be retained using new features, and the tests with different decision rules will indicate one for which the performance of recognition is satisfactory.

REFERENCES

1. Dobrin, M. B., Introduction to Geophysical Prospecting, Second Edition, McGraw Hill Company, Inc., 1960.
2. Russell, W. L., Principles of Petroleum Geology, Second Edition, McGraw Hill Company, Inc., 1960.
3. Kulinkovich, A. Ye. et al., "Utilization of Digital Computers to Distinguish Boundaries of Beds and Identify Sandstones from Electrical Log Data," International Geology Review, Vol. 8, No. 4, April, 1966, pp. 416-420.
4. Jeffries, F. S., "A Self-Thought Computer Program to Interpret Micrologs," The Log Analyst, Vol. 7, No. 3, July, 1967, pp. 15-18.
5. Bongard, M. M. et.al., "Utilization of a Learning-type Computer Program for Identification of Oil-bearing Layers," Academy of Sciences of the USSR, Siberian Branch, Geology and Geophysics, No. 6, 1966, pp. 96-105, (in Russian).
6. Guberman, Sh. A. et al., "Use of an Image-Recognition Algorithm to Solve Problems in Applied Geophysics," Doklady of the Academy of Sciences USSR, Vol. 154, No. 5, pp. 12-13, January, 1964, Earth Science Section (translation published by the American Geological Institute).
7. Guberman, Sh. A. et al., "Evaluation of Information Content of the Geophysical Parameters in the Separation of Oil-bearing Layers through Learning-type Programs," Izvestiya Academy of Sciences USSR, Physics of the Solid Earth, No. 12, 1966, pp. 777-781 (in English).
8. Kontorovich, A. E. et al., "Application of Digital Computers in Predicting Oil Quality," Doklady of the Academy of Sciences USSR, Vol. 180, No. 2, 1968, pp. 425-427, (in Russian).
9. Krumbein, W. C. and Graybill, F. A., An Introduction to Statistical Models in Geology, McGraw Hill Company, Inc., New York, 1965.
10. Petrashen, G. I., "Application of Mathematical and Machine-Computing Methods to Certain Problems in Geology and Geophysics," International Geology Review, Vol. 10, No. 4, April, 1968, pp. 478-484.
11. Harbaugh, J. W. and Merriam, D. F., Computer Applications in Stratigraphic Analysis, John Wiley and Sons, Inc., New York, 1968.
12. Howd, F. H., "The Taxonomy Program--A Computer Technique for Classifying Geologic Data," Quarterly of the Colorado School of Mines, Vol. 59, No. 4, October, 1964, pp. 207-222.
13. Polkanov, V. P. et al., "Detection of Local Uplifts from Physiographic Indices Using Digital Computers," Doklady of the Academy of Sciences USSR, Vol. 167, No. 3, Earth Science Section (published by the American Geological Institute), March, 1966, pp. 28-30, (in English).

14. Merriam, D. F., and Sneath, R. H. A., "Quantitative Comparison of Contour Maps," Journal of Geophysical Research, Vol. 71, No. 4, February, 1966, pp. 1105-1115.
15. Loudon, T. V., "The Use of Eigenvector Methods in Describing Surfaces," in Computer Applications in the Earth Sciences: Colloquium on Trend Analysis, edited by Daniel F. Merriam and Nan Carnahan Cocke, Computer Contribution 19, State Geological Survey, The University of Kansas, Lawrence, 1967, pp. 12-15.
16. Man, C. J., "Spectral-Density Analysis of Stratigraphic Data," in Computer Applications in the Earth Sciences: Colloquium on Time-Series Analysis, edited by Daniel F. Merriam, Computer Contribution 18, State Geological Survey, The University of Kansas, Lawrence, 1967.
17. Potter, P. E., and Blakely, R. F., "Random Processes and Lithologic Transitions," Journal of Geology, Vol. 76, No. 2, March, 1968, pp. 154-170.
18. Nezhenskiy, I. A., "Use of Discriminant Functions in Describing Types of Ore Deposits of Similar Mineral Composition," Doklady of the Academy of Sciences USSR, Vol. 174, No. 6, Earth Science Section, (translation published by the American Geological Institute), 1967, pp. 102-104.
19. Dobretsov, N. L. et.al., "Utilization of Multivariate Statistical Analysis in Solving the Prediction Problems on the Example of Diamond-bearing Kimberlites," Academy of Sciences of the USSR, Siberian Branch, Geology and Geophysics, No. 4, 1967, pp. 70-75, (in Russian).
20. Voronin, Yu. A. and Karataeva, G. N., "On a Possible Method of Defining a Holotype and Its Application in Solving the Problems of Diagnosis (Recognition)," Academy of Sciences of the USSR, Siberian Branch, Geology and Geophysics, No. 4, 1967, pp. 70-75, (in Russian).
21. Voronin, Yu. A., "On Investigations Related to the Applications of Mathematical Methods and Digital Computers in Geology," Academy of Sciences of the USSR, Siberian Branch, Geology and Geophysics, No. 10, 1967, pp. 75-81, (in Russian).
22. Yelkina, V. N. et. al., "On a Possibility of Application of Pattern Recognition Methods in Paleontology," Academy of Sciences of the USSR, Siberian Branch, Geology and Geophysics, No. 9, 1967, pp. 75-79, (in Russian).
23. Gower, J. C., "A Comparison of Some Methods of Cluster Analysis," Biometrics, Vol. 23, No. 4, December, 1967, pp. 623-637.
24. Keylis-Borok, V. I., "Use of Electronic Computers in Study of Structure and Development of the Earth," International Geology Review, Vol. 9, No. 10, October, 1967, pp. 1284-1289.

25. Naimark, B. M., "An Algorithm for Automatic Recognition of a Seismic Signal by a Computer," Reviews of Geophysics, Vol. 3, No. 1, February, 1965, pp. 240-246.
26. Zidarov, D., "Solution of Some Inverse Problems of Applied Geophysics," Geophysical Prospecting, Vol. 13, No. 2, June, 1965, pp. 240-246.
27. Rastrigin, L. A., "Solution of Inverse Problems by Statistical Optimization Methods," Review of Geophysics, Vol. 3, No. 1, February, 1965, pp. 111-114.
28. Levin, F. K., "Control and Identification Problems in Exploration Geophysics," Presented at the 1968 Joint Automatic Control Conference, University of Michigan, Ann Arbor, Michigan.
29. Asbel, I. J. et al., "A Technique of a Joint Interpretation of Travel-Time and Amplitude-Distance Curves in Upper Mantle Studies," The Geophysical Journal of the Royal Astronomical Society, Vol. 11, September, 1966, pp. 25-55.
30. Lyubimova, E. A., "Use of Thermal Data for Decreasing the Ambiguity of Inverse Problems in Seismology," Doklady of the Academy of Sciences USSR, Vol. 179, No. 1, 1968, pp. 80-83, (in Russian).
31. Robinson, E. A., "Predictive Decomposition of Time Series with Application to Seismic Exploration," Geophysics, Vol. 32, No. 3, June, 1967, pp. 418-484.
32. Lacoss, R. T., "Adaptive Combining of Wideband Array Data for Optimal Reception," IEEE Transaction on Geoscience Electronics, GE-6, No. 2, May, 1968, pp. 78-96.
33. Nikitin, A. A., "Use of Statistical Theory of Signal Detection to Reveal Weak Geophysical Anomalies," International Geology Review, Vol. 10, No. 6, June, 1968, pp. 672-679.
34. Nagy, G., "State of the Art in Pattern Recognition," Proceedings of the IEEE, Vol. 56, No. 5, May, 1968, pp. 836-862.
35. Photiadi, E. E., et. al., "The Experiments with the Construction of Algorithms and Computer Programs for Processing of Geological Data," Academy of Sciences of the USSR, Siberian Branch, Geology and Geophysics, No. 10, 1966, pp. 10-16, (in Russian).

COMPUTER-ASSISTED FINGERPRINT ENCODING AND IDENTIFICATION

C. N. LIU
IBM and Purdue University
School of Electrical Engineering
Purdue University, Lafayette, Indiana

ABSTRACT

Fingerprints have been used systematically as a means of establishing personal identification for nearly a hundred years. The ten-finger file has been accepted by almost every country to be the single most reliable and convenient way of general personal identification. However, as population grows, the task of searching fingerprint files becomes increasingly more difficult. There has been a growing interest in the possibility of using a computer to automate the fingerprint searching process. Several feasibility studies have been made. It is generally concluded that the task of computerizing file searches of ten-finger files on the basis of existing gross typings of ten fingers is not too forbidding. Several experimental systems are being developed. However, the identification of a single latent fingerprint by a computer remains to be an unsolved and difficult problem.

It has been suggested that the minute ridge characteristics of a finger may be used as a basis for automatic identification of single fingers. Each finger in the file is represented by its minutiae patterns. Identification is achieved by matching the minutiae patterns. This is, in fact, what fingerprint experts do when they are asked to verify the identities of latent prints. The major difficulty in attempting to computerize this process lies in the task of automatic recognition of ridge characteristics. This paper presents a man-machine system for encoding fingerprints in terms of ridge characteristics. Experimental results on the usefulness and efficiency of the proposed system are reported. Experiments of matching multiple impressions of 53 fingers against a file of 102 fingers are described.

PATTERN CLASSIFICATION OF DATA USING A REINFORCEMENT LEARNING SCHEME

R. W. McLAREN, G. C. BRESHEARS, A. C. PAN
Dept. of Elect. Engr., University of Missouri, Columbia, Missouri

ABSTRACT

This paper concerns the application of a reinforcement learning algorithm to two different pattern classification decision structures which are suitable for the on-line classification of particular types of patterns. One of the decision structures, called the randomized direct decision scheme, is defined on feature space; it consists of on-line generation of a sequence of spherical regions. A set of decisions representing the pattern classes is assigned to each sphere as it is generated. The algorithm is applied iteratively to the applicable sphere to learn the "best" decision (lowest error) for each sphere while causing the average error to decrease monotonically toward a minimum level. Simulation results are presented to demonstrate the applicability of the scheme. The second approach, called a randomized decision scheme defined on an automaton, consists of first defining an automaton structure which provides a particular input/output relationship. Then, with the automaton receiving a sequence of integer-valued feature vectors, the algorithm is applied to a set of decisions assigned to each state transition of the automaton to learn the "best" decision for each transition and decrease the average error monotonically toward a minimum level. The two decision structures are particularly applicable to pattern classes that are only locally clustered but scattered in feature space. The need for specific decision boundaries is eliminated.

INTRODUCTION

In recent years, pattern classification or recognition has been applied to an ever increasing variety of data patterns. Examples of such patterns include hand-printed or written characters, speech, weather prediction, and medical data such as EKG waveforms, disease data, and roentgenographic signs. The pattern classification system performs measurements and preprocessing on the input pattern sample data to form features and then applies a classification or decision scheme to the features in order to estimate or select a decision as to the pattern class of the sample. From an engineering viewpoint, the basic problem of pattern recognition is to design the pattern classification system for a specific class of data patterns to provide a minimum average percent misrecognition, or at least below a specified level. If erroneous decisions are not all equally weighted, then a "loss" function can be substituted for the percent misrecognition. The design includes: (1) selecting the measurements and features and (2) selecting a decision scheme. This paper will concern only the latter problem. Decision or pattern classification procedures are many and varied [1] - [10]. The selection and manner of application of a particular decision or classification scheme depends on (1) the nature of the pattern classes in terms of the measurements and chosen features and (2) the amount of apriori knowledge and/or assumptions available concerning the patterns. Here, so-called "noisy patterns" rather than fixed patterns will be emphasized. If the patterns are fixed or well-defined by known characteristics or assumptions, the pattern classifier can operate effectively without interaction with an environment. If patterns are to be predicted, as in weather prediction, or if external training is required, the pattern classifier will operate effectively by interaction with its environment. Parametric decision schemes, [1] [2], are applica-

ble to pattern classes described by a definite set of parameters such as mean values, mode locations, etc., which are either known or can be estimated. If the estimation of unknown pattern parameters is not possible or practical, then non-parametric decision schemes should be used, [3], [4], such as those using metric properties of the patterns and/or using a training procedure coupled with a hyperplane technique [5]. It is desired that the a priori knowledge and/or assumptions concerning the metric properties of the patterns in feature space such as clustering, mode locations, convexity, separability, etc. be utilized to determine the optimal form and location of the decision boundaries which partition feature space. If the pattern class parameters drift in time, however, a pattern classification scheme with adaptive properties can maintain acceptable performance if sufficient a priori information is available regarding the pattern classes. If such knowledge (or assumptions) is lacking, a pattern classification system with learning properties can be used to achieve long-term improvement in the performance. Learning properties are incorporated into the decision scheme for (1) parametric learning, such as learning pattern class statistics [2], (2) non-parametric learning such as using an external trainer to adjust a set of hyperplane decision boundaries [6], or (3) learning using the system's decision. Whether a particular pattern classification system learns with [6] or [1] or without [7] an external supervisor (or "teacher") is often a source of confusion. This confusion stems partly from various interpretations of exactly what comprises supervision. Perfect training (without error) certainly represents supervision; however, if the training is "noisy" or if the system is trained (without error) for only an initial period, then the role of supervision is not so clear. If the pattern classification system interacts with its environment so that it receives information from it useful for evaluating performance, the role of direct supervision is weakened. Here such interaction will not be considered supervision. Learning without such environmental interaction requires sufficient a priori knowledge or assumptions such as distributions, etc. Pattern classification schemes can learn on-line as the classification is performed or learn off-line, in which the learning (training) and operating (no learning) phases are separated. This paper concerns the application of a reinforcement learning algorithm to two different pattern classification decision structures for the on-line classification of particular types of patterns.

THE PATTERN CLASSIFICATION PROBLEM

A pattern classification system which interacts with its environment can be viewed as shown in Fig. 1. It consists of two main sections, the receptor and the decision process. For input pattern sample S , the receptor generates a set of measurements $G = [g_1 \dots g_c]$ and then extracts a "good" set of features $U = [u_1 \dots u_d]$ from the measurements. Although the problem of selecting both a "good" set of measurements and a "good" set of features is important, it will not be considered here. The natures of both the environment and the pattern classes influence the choice of both the decision and learning schemes for the pattern classification system. Feature vector U , in general, can assume a continuum of values in feature space E_s , an s -dimensional Euclidean space. Let the pattern classes be denoted by θ_i , $i = 1, 2, \dots, N$, finite N . A large number of pattern classification schemes are based on the concept or assumption that if two different pattern samples U_1 and U_2 are "close together" in some sense, then they must both be samples from the same pattern class. Applying this concept to feature space results in the creation of decision boundaries that partition feature space into decision regions, the basic concept now being applied to each region. If these boundaries are not predetermined for the classification process,

then they must be generated or formed through training, estimation procedures, or other self-adjusting procedures of the pattern classifier. The classification of a pattern sample in feature space is then determined by its location in reference to these boundaries. If the pattern samples possess only local clustering properties in which even a small number of pattern classes generate a large number of local clusters of various classes scattered throughout feature space, the formation of decision boundaries becomes difficult, if not impractical, particularly in on-line classification procedures.

This paper concerns two decision structures, each using the features in a different manner. With such structures, the need for decision boundaries, such as hyperplanes, is eliminated. In addition, the decision selected for any given pattern sample will be independent of such boundaries. Arbitrarily small regions will be formed in feature space by the structures, but they will be fixed and not dependent on the (training) samples. Such decision structures are applicable to data patterns which are scattered into local clusters, such as some types of medical data. These two decision structures will incorporate learning properties. The pattern classifier uses environmental information or feedback in an on-line manner to reduce the average loss toward a minimum value while learning the optimum decision for each arbitrarily small region or point in feature space. Incorporating the learning process into the decision structure is accomplished through the iterative application of a reinforcement learning algorithm. The basic algorithm is applied in two different ways to the two different decision structures proposed here. In a later section the simulation results of the application of these two decision structures to particular data patterns will be presented and discussed. The learning algorithm will be described in the next section.

THE REINFORCEMENT LEARNING ALGORITHM

Consider a learning system with an output which can assume one of the alternatives or responses $r_1, r_2, \dots, r_N$. Let $r(k)$ be the output response at step k , $k = 0, 1, 2, \dots$, in a discrete time learning process. Let $p_i(k) = \Pr\{r(k)=r_i\}$, $i=1, \dots, N$. Let the learning system interact with a stochastic environment in such a way that, during successive time intervals, as a "reaction" to the response $r(k)$, the environment provides feedback information $f(k)$ which can be utilized by the learning system to calculate a normalized performance measure $Z(k)$, $0 \leq Z(k) \leq 1$. A lower value of $Z(k)$ is interpreted by the learning system to be more desirable. Let the conditional p.d.f. $p(Z|r_i)$ be stationary for any given i , but generally unknown to the designer. In addition, let $m_i = E_Z(Z|r_i)$, all i , where m_i is the conditional expected performance measure given response r_i . Assuming that all m_i are defined but unknown, it is the goal of the learning system to learn the response which corresponds to the smallest m_i . To accomplish this, let $r(k) = r_j$ and define the following reinforcement learning algorithm.

$$\begin{aligned} p_j(k+1) &= \alpha p_j(k) + (1 - \alpha)\{1 - Z(k)\}, \\ p_i(k+1) &= \alpha p_i(k) + (1 - \alpha)\{Z(k)/(N-1)\}, \quad i \neq j, \end{aligned} \quad (1)$$

all i , where α is a learning parameter, $1/N < \alpha < 1$. The properties of this algorithm are considered in [11] and are discussed in detail in [12]. It is sufficient here only to outline the properties useful for purposes here. Under conditions such as the monotonicity of the expected $p_i(k)$'s

and $\Delta p_i(k)$'s and letting $p_i(0) = 1/N$, all i , the expected values of the $p_i(k)$'s will converge monotonically toward final values while the expected value of $Z(k)$ will converge monotonically toward a minimum value. The final values of the $E\{p_i(k)\}$'s are in reverse magnitude order as the corresponding conditional expected values of $Z(k)$, i.e., the m_i 's. Specifically, this means that in the limit as $k \rightarrow \infty$, the response giving the smallest m_i will have the highest (expected) probability. This is the basis for selecting the optimal response. Hence, this describes the on-line learning behavior of the system as it learns the "best" response. In addition, as a measure for comparing or describing the convergence behavior of the learning system, a learning rate is defined as the initial change in expected performance. This rate decreases as (1) α increases, (2) the variance of $Z(k)$ increases, (3) N increases, or (4) as the m_i values become closer together [12]. A low rate of convergence can cause the expected performance to differ little from the initial value, even after a long time. The rate of convergence can be increased while lowering the expected performance level by periodically eliminating responses having low expected probabilities (estimated by the time averages) and continuing the learning process with fewer responses each time. This algorithm can be generated by a stochastic automaton having input $Z(k)$ and states $Q_1, \dots, Q_N$, each state being associated with one response $r_1, \dots, r_N$. This automaton exists in one state during any step and undergoes random state transitions from step to step [13]. In [14], this algorithm was applied to select the optimal positions of a set of decision hyperplanes partitioning feature space. In the following sections, the application of the reinforcement learning algorithm to two different decision structures that do not use such decision boundaries will be discussed.

A RANDOMIZED DIRECT DECISION SCHEME DEFINED ON FEATURE SPACE

In this section, the reinforcement learning algorithm defined by Eq.(1) is applied to a particular decision structure for pattern classification. Basically, this involves assigning a learning system (described previously) to each arbitrarily small region in feature space that contains one or more pattern samples. Such a scheme is particularly applicable to data in which one or more pattern classes consists of scattered small clusters in feature space. Consider an on-line learning pattern classification scheme as follows. Let $U(k)$ be a sample point in feature space at step k . Let $Z(k)$ as described in the previous section be available or calculable from environmental feedback. Let lower values of $Z(k)$ be more desirable. Let $U(0)$ be the center of a sphere S_0 with radius T . A sequence of spheres $S_0, S_1, \dots$, will be generated. When a new sphere S_i is generated, a learning system as described in the previous section is assigned to it; its input is $Z(k)$ and its responses are the decisions $d_{i1}, d_{i2}, \dots, d_{iN}$, where d_{ij} refers to deciding pattern class C_j (for any i). For any sphere, $p_i(0) = 1/N$, all i . At any step k in the learning process, only one sphere will apply; the decision probabilities of that sphere are changed according to Eq.(1) using $Z(k)$, while those of all other spheres remain unchanged. After Eq.(1) is applied to sphere S_0 , using $Z(0)$, a new pattern sample is taken. If $U(1)$ is in sphere S_0 , the algorithm is repeated, using $Z(1)$. If $U(1)$ is outside of S_0 , then a new sphere is formed having center at $U(1)$ and radius T . The algorithm assigned to this sphere is applied (for the first time) according to Eq.(1) and using $Z(1)$. Let this generation process be repeated step by step until spheres $S_0, S_1, \dots, S_{t-1}$ have been generated up to step $k-1$. One of three possibilities may occur for $U(k)$. (1) If $U(k)$ falls in only one of the existing spheres S_i , then the learning algorithm associated with sphere S_i is applied to the decisions for this sphere, using $Z(k)$. Decision probabilities for all other spheres remain unchanged. (2) If $U(k)$ falls

outside of all existing spheres, a new sphere S_t is formed with center $U(k)$ and radius T ; a learning algorithm defined by Eq.(1) is applied to the N decisions for this sphere for the first time, using $Z(k)$. Other spheres are unaffected. (3) If $U(k)$ falls inside two or more (overlapping) existing spheres, the sphere whose center is closest (Euclidean distance) to $U(k)$ is assigned this point, no new sphere being formed. Using $Z(k)$, the learning algorithm is applied to this sphere, all other spheres being unaffected. At any step k , only one sphere changes its decision probabilities according to the learning algorithm, while the decision probabilities for all other spheres remain the same. Therefore, because the expected performance measure for each sphere converges monotonically toward a minimum value, the overall expected performance measure will converge monotonically toward a minimum value. This can be seen more easily as follows. Let $b_j(k)$ be the probability that $U(k)$ falls in (or is assigned to) sphere S_j , having expected decision probabilities $E_{Z\{p_{ij}(k)\}}$, $j = 1, \dots, N$. The expected (overall) performance is then given by

$$P_m(k) = \sum_{j=0}^t b_j(k) \left\{ \sum_{i=1}^N E_{Z\{p_{ij}(k)\}} m_i \right\} \quad (2)$$

where $p_{ij}(k)$ is the probability of the j^{th} decision for sphere S_j at step k . From step to step, only one of the inner sums decreases monotonically, all others remaining unchanged. Hence, the overall expected performance decreases monotonically (toward a minimum). In addition, as more and more samples fall in (or are assigned to) a given sphere, the "best" decision for that sphere will gain the highest expected probability. After the best decision has been selected for each sphere, the randomized decision rule will be replaced by a fixed one.

Simulation Results

A high-speed digital computer was employed to simulate the application of this scheme to three different data classification problems. The scheme was first applied to a collection of 1100 samples each of three hand-printed English characters from which 4 features were extracted, each representing the number of intersections of a line with the sample character (placed in a standard area as in [15]). The features are therefore non-negative integers. The performance measure used was the time average performance defined by,

$$\bar{P}_m(k) = \frac{1}{k+1} \sum_{\ell=0}^k Z(\ell) \quad , \quad (3)$$

where $Z(k) = 0$ for no error and $Z(k) = 1$ for error was used (more general forms for $Z(k)$ were also used). Initial results showed that the radius T had an effect on the terminal level of $\bar{P}_m(k)$. An optimum value (lowest $\bar{P}_m(k)$) of $T = 2.2$ was used for subsequent work ($T = 0$ is excluded). Due to the nature of $Z(k)$, convergence was rapid; this resulted in learning parameter α having little effect on performance. Using about 3400 samples presented randomly for the on-line learning process, the terminal average error was about 17 %.

As a variation on the basic scheme, when the number of pattern samples falling in (or assigned to) reaches a prescribed number, the decision with the highest time average probability is assigned to that sphere for all future samples falling in (or assigned to) that sphere. This is done on-line as the classifier learns; an increasing number of spheres become associated with one decision. This reduces the average error. The simulation results for this scheme show that the terminal time average error (perform-

ance) was about 14 %. It is noted that learning continues for a long time because much time is required for many of the spheres to attain the prescribed number of samples.

Another variation on the basic scheme consisted of splitting the operation of the learning pattern classifier into an initial on-line training period and a non-learning "operating" period. In a particular simulation example, about 1700 random samples were used to train the decision structure using the same basic on-line technique as described initially in this main section. Then, for each of the spheres generated by this training, the time average decision probabilities are used to select the "best" decision (that having the highest such probability) and then assigning it to that sphere. In the simulation results presented here, the time average quantities are used as estimates for the expected values. Then, with one decision assigned to each sphere, $\tilde{P}(k)$ is reinitialized and more pattern samples are processed (no new spheres are formed). After many iterations without learning, the time average error leveled at about 8.3 %. Zero error is not expected because in some cases different character class samples had the same feature values.

The decision scheme described above was also applied to another set of data patterns through simulation. This data consisted again of hand-printed English characters in the form of the 8 features described in [15]. These features may assume a continuum of values. A preliminary study showed that the optimum (lowest average error) sphere radius was about $T=2.2$. Then, the scheme described in the previous paragraph in which operation is split up into training and operating was applied to this data. The results are summarized in Table I below.

Table I Simulation Results for Characters Described by 8 Features

No. of Characters	No. of Training Samples	Terminal Time Average Error
3	800	21%
6	800	37%
6	2500	36%
10	2500	31%
26	6000	50%

The number of characters refers to N , the number of pattern classes for a given simulation run. The training samples presented to the classifier were randomly selected from a smaller number of data samples. Different combinations of N characters gave different results for the third column; hence, this column represents "typical results", some combinations giving a lower time average error due to their greater separability. This accounts for the lower time average error for $N = 10$ than for $N = 6$; this result shows the effect of the chosen features. A large increase in training samples often yielded only a small improvement in error.

This same decision scheme was applied to some particular medical data. This data represented roentgenographic signs arising from an x-ray; this data will be described in a later section along with some of the difficulties encountered in using it. These difficulties prevented useful results; however, the preliminary results were encouraging and indicated the directions for further investigation.

Variations and modifications could be applied to this scheme to improve performance, such as using weighted distances or a variable radius. The data used was conveniently available; this is particularly meaningful for medical data, which is not so easily obtained. However, the application of

a reinforcement learning algorithm to a particular decision structure has been presented; simulation results for this scheme demonstrate its successful application. The scheme is particularly applicable to data patterns which are not easily separated by decision boundaries.

A RANDOMIZED DECISION SCHEME DEFINED ON AN AUTOMATON STRUCTURE

This section considers the application of the reinforcement learning algorithm to a second decision structure for on-line pattern classification; this method is applicable to discrete-valued data. Consider an automaton A operating sequentially in time; let $q(k)$, $x(k)$, and $y(k)$ denote the internal state, input, and output for A, respectively, at step k . $q(0)$ may or may not be specified. Let Q , X , and Y represent the set of states, input alphabet, and output alphabet of A, respectively; all are finite. Input tape $x(0)x(1)...x(k)$ causes the automaton to pass from state to state giving the output sequence $y(0)y(1)...y(k)$. A stochastic automaton structure as described in [16] is to be defined on A to yield a given input/output relationship which can be adapted for pattern recognition. The desired class of relationships is described in [16]. Briefly, set,

$$y(k) = h\{x(k), x(k-1), \dots, x(k-w)\}, \quad (4)$$

where $k > w$, w is a finite non-negative integer, $w = 0, 1, \dots, B$, and $x(k)$, $x(k-1), \dots, x(k-w)$ are any set of symbols from X and $y(k)$ is in Y . h is a deterministic mapping from the space of sets of $w+1$ input symbols into Y ; h does not depend on k . Let h be defined by the set of relationships,

$$y_e = h\{x(k) = x_{t_k}, k=0, 1, \dots, w\} \quad (5)$$

where the x_{t_k} are any set of $w+1$ symbols $x \in X$ and y is any $y \in Y$. With p distinct input t_k and n distinct output symbols, in general, there would be p^{w+1} relationships of the form of Eq.(5) specified; n to the p^{w+1} power such sets of relationships are possible. To achieve a particular I/O relationship, let the automaton structure be initialized as a stochastic automaton with X and Y as input and output alphabets as before and having states $q_1, \dots, q_M$, where $M = p^w$. Let q_i be defined by, $q_i = (v_1, v_2, \dots, v_w)$, where the v_j are any $x \in X$; there are $M = p^w$ distinct sets of $(v_1, \dots, v_w)$, each defining a state of the automaton. Define the state transition function as,

$$q(k+1) = (v_2, v_3, \dots, v_w, x_i) = f_s\{q(k)=(v_1, \dots, v_w), x(k)=x_i\}, \quad (6)$$

where $v_1, \dots, v_w$ and $x_i \in X$. If the initial state probability distribution is uniform over $p^w M$ states, it remains uniform under the transformation defined by Eq.(6) [16]. For the automaton, the state transition labeled x_i/y_i occurs when the input is x_i and then gives y_i as output. There are only p^{w+1} allowed transitions which is also the number of distinct sets of $w+1$ input symbols in X . Hence, associate each state transition with one such set. The output function g defined over the $M = p^w$ states is not needed.

To incorporate learning in this automaton structure, let a reinforcement learning algorithm as previously described be applied to each state/input symbol pair, or equivalently, to each transition. If $q(k) = q_i$ and $x(k) = x_i$, then the i - j learning algorithm is applied to the response probabilities of the x_i/Y transition, $r_i = y_i$, all i , using an appropriate $Z(k)$. The response probabilities of all other transitions remain unchanged. If small $Z(k)$ values are desirable, then using reasoning similar to that concerning Eq.(2), wherein $b_j(k)$ is to be the state probability, the overall expected

performance measure will converge monotonically toward a minimum level while for each transition, the "best" response will eventually have the highest average probability. If $Z(k)$ is a measure for how "close" the actual output symbol is to the desired one, then this learning decision structure will learn the desired I/O relationship while lowering the expected performance.

In order to apply this learning decision structure to pattern classification, let the input data to the learning automaton be represented as a sequence of digits. Such a sequence could consist of a sequence of integer-valued feature vectors of dimension d , possibly in higher dimensional binary form. Let the feature vector $U = (u_1 \dots u_d)$ be a word consisting of symbols from alphabet X of automaton A . The input tape consists of the sequence of words, $\{x(0)x(1)\dots x(d-1)\}\{x(d)\dots x(2d-1)\} \dots \{x(Kd)x(Kd+1)\dots x[(K+1)d-1]\}$. The output of A is a sequence of symbols from the alphabet $Y = \{1, 2, \dots, N\}$, where N is the number of pattern classes. Let the output represent the decision or pattern classification; in this sequence, only every d symbol is meaningful as a decision because d steps are required for the automaton to receive a full input word (feature vector) before yielding a decision. Hence, a value of $Z(k)$ was provided only every d steps, only at which times, the learning algorithm was applied to this decision structure in the manner discussed. The result is to learn the best decision while minimizing the expected $Z(k)$, which is an error function here.

Simulation Results

The application of the above described decision structure to the on-line classification of particular data patterns was simulated. The same data, consisting of 4 features extracted from about 1100 samples each of 3 English characters, was used; this data is integer-valued. A 256 state automaton as described previously was first "trained" in an on-line manner, using 5000 input samples randomly selected from about 3400 samples representing the 3 equally probable characters. The resulting time average performance measure as a function of time k is shown in Fig. 2; for this particular result, $Z(k) = 0$ for a correct classification and $Z(k) = 1$ for an incorrect one. After the training period, the best decision or response was selected for each transition by determining the one with the highest time average probability. Assigning the best decision to each transition and resetting the time average error $\bar{P}_m(k)$ to 0, the automaton continued to receive feature vectors. After many iterations, the time average error tended toward a level of 5%. These results demonstrate the applicability of the proposed decision structure to certain classes of data patterns; no decision boundaries are needed.

An attempt was also made to apply this learning decision scheme to a particular set of medical data. Although the difficulties encountered precluded acceptable results, the preliminary results were encouraging.

Prediction

The stochastic automaton decision structure described in the previous section is also applicable to pattern prediction. Each transition of the automaton corresponds to a finite sequence of input symbols. At a given step the transition which occurs represents the prediction of the next input (or inputs) in the input sequence. A comparison of predicted and actual symbols is used to form $Z(k)$; $Z(k)$ is used in the learning algorithm to alter the decision probabilities for this transition. No other transition is effected. This scheme was simulated for a quantized form of a simple repeated waveform. Results showed that the automaton provided a waveform which approached the desired one while lowering the average performance.

Some particular medical data patterns were available for classification by the two decision schemes discussed previously. This data consists of about 1000 pattern samples, each consisting of a 28 dimensional feature vector; each dimension represents an integer-valued roentgenographic sign taken from a chest x-ray of a patient. A pattern sample is to be classified into one of 14 classes of heart disease. This data was provided by the Department of Radiology of the University of Missouri. The large number of dimensions and pattern classes presented problems for both of the pattern classification schemes using the learning algorithm. In the direct decision scheme, the large number of dimensions and pattern classes slowed the learning rate a great deal, resulting in poor initial performance. An additional difficulty arose because in some dimensions, a "0" referred to numerical 0 while in others, it meant that no measurement was taken. In addition to these difficulties for the second decision scheme, the high dimensionality prevented the use of one automaton due to storage. It was necessary to break-up the 28 dimensions into subsets, each being the input to a separate automaton. The resulting automata operate in parallel, requiring less storage. Each automaton is trained on a subset of the 28 dimensions, while the time average performance was averaged over all automata. After training, the selection of the "best" decision for a given transition was based on the time average probabilities, averaged over the corresponding transitions of all automata. Another difficulty encountered with this scheme is due to the relative importance of some subset of features in separating a given pair of pattern classes, the subsets being different for different pairs. These problems yielded results which were encouraging but not impressive.

CONCLUSIONS

Two approaches to the pattern classification of data using an on-line reinforcement learning algorithm have been presented. In the first approach the algorithm is successfully applied to select the best decision for each arbitrarily small region in feature space where pattern samples occur, while lowering the average performance measure as the system operates. Computer simulation results using different data patterns demonstrated the applicability of the scheme. In the second approach, the algorithm was successfully applied to select the best decision for each transition in an automaton, while lowering the average performance as the system operates. Computer simulation results for this scheme verified the applicability of the scheme. These two schemes or decision structures are particularly applicable to data patterns which are "scattered" into local clusters in feature space; no decision boundaries are needed. Preliminary results with medical data indicate the problems involved and show the direction for further work.

REFERENCES

1. Abramson, N., and Braverman, D., "Learning to Recognize Patterns in a Random Environment," IEEE Trans. on Infor. Th., vol. 8, no. 5, 1962.
2. Keehn, D. G., Learning the Mean Vector and Covariance Matrix of Gaussian Signals in Pattern Recognition, Stanford Rept. 2003-6, Feb., 1963.
3. Whitney, A. W., and Dwyer, S. J., "Performance and Implementation of the K-Nearest Neighbor Decision Rule with Incorrectly Identified Training Samples," Proc. of 4th Annual Allerton Conf. on Cir. and Sys. Theory, University of Illinois, 1966.
4. Sebestyen, G. S., and Edie, J., "An Algorithm for Non-parametric Pattern Recognition," IEEE Trans. on Elect. Comp., vol. 15, no. 6, 1966.
5. Highleyman, W. H., "Linear Decision Functions with Application to Pattern Recognition," Proc. IRE, vol. 50, no. 6, pp. 1501-1514, 1962.

6. Nilsson, N. J., Learning Machines, McGraw-Hill, New York, 1965.
7. Spragins, J., "Learning Without a Teacher," IEEE Trans. on Infor. Th., vol. IT-12, no. 2, pp. 223-230, 1966.
8. Sebestyen, G. S., Decision Making Processes in Pattern Recognition, Macmillan Co., New York, 1962.
9. Fralick, S. C., "Learning to Recognize Patterns Without a Teacher," IEEE Trans. on Infor. Theory, vol. IT-13, pp. 57-64, 1967.
10. Chen, C. H., "A Note on Sequential Decision Approach to Pattern Recognition and Machine Learning," Infor. and Control, vol. 9, no. 6, 1966.
11. McLaren, R. W., "A Stochastic Automaton Model for the Synthesis of Learning Systems," IEEE Trans. on Sys. Sci. & Cyber., vol. 2, no. 2, '66.
12. Fu, K.-S., and McLaren, R. W., "An Application of Stochastic Automata to the Synthesis of Learning Systems," Purdue Univ. Tech. Rept. No. EE65-17, Lafayette, Indiana, Sept., 1965.
13. Fu, K.-S., "Stochastic Automata as Models of Learning Systems," in Computers and Information Sciences-II (J. T. Tou, Ed.), Academic Press, New York, pp. 177-191, 1967.
14. McLaren, R. W., "The Application of a Stochastic Automaton Model to Pattern Recognition," Proc. of Hawaii Inter. Conf. on Sys. Sci., 1968.
15. Marill, T., and Green, D. M., "Statistical Recognition Functions and the Design of Pattern Recognizers," IRE Trans. on Elect. Comp., vol. EC-9, no. 4, Dec., 1960.
16. McLaren, R. W., "A Technique for Generating the Structure of an Automaton to Satisfy a Given Input/Output Relationship," Proc. of the Second Annual Princeton Conf. on Information Sciences and Systems, 1968.

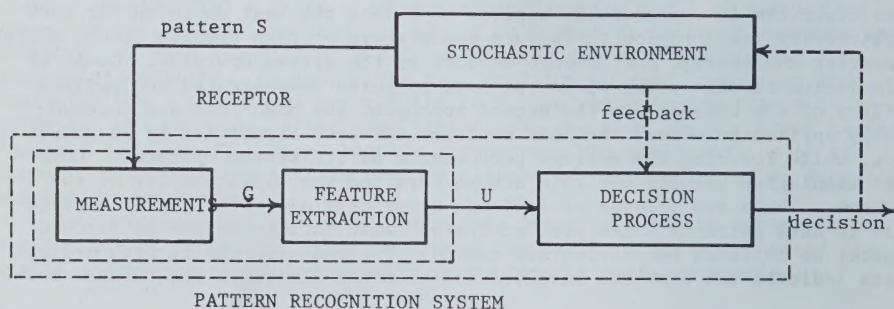


Fig. 1 Pattern recognition system interacting with its environment.

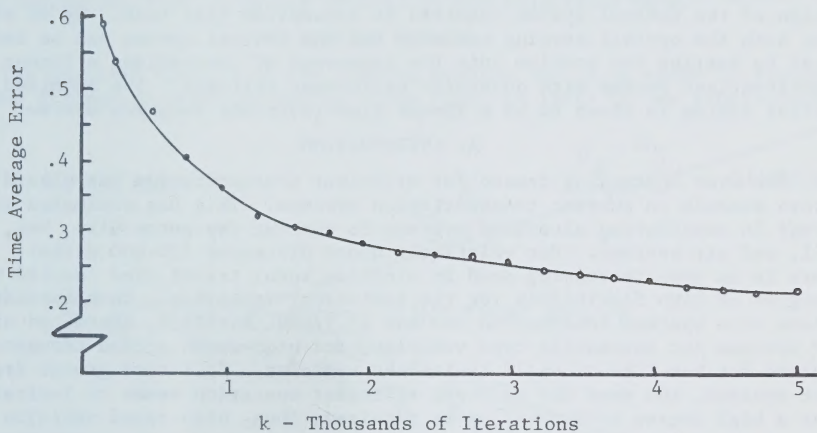


Fig. 2 Time average performance vs. k for a 256 state automaton

MICHAEL ATHANS

Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02142

ABSTRACT

This paper considers the problem of deciding the best way of merging two strings of high-speed vehicles into a single guideway as well as the design of the control system required to accomplish this task. It is shown that both the optimal merging sequence and the control system can be determined by casting the problem into the framework of controlling a linear time-invariant system with quadratic performance criteria. The required control system is shown to be a linear time-invariant feedback system.

1. INTRODUCTION

The ever increasing demand for efficient transportation has placed severe demands on current transportation systems. This has motivated interest in considering alternate systems to current day automobile, bus, rail, and air systems. For relatively short distances (50-500 miles) there is an ever increasing need to minimize total travel time coupled with a degree of high flexibility for the individual traveller. Such considerations have sparked interest in the use of V/STOL aircraft, automated highway systems for automobile type vehicles, and high-speed ground transport systems for bus-like or small train-like vehicles. In future ground transport systems, the need for safe and efficient operation seems to indicate that a high degree of automation is required; thus, high-speed vehicles under computer control, requiring little or no human control, merit investigation.

For the past three years a considerable research effort has been focused upon certain control problems arising in automated ground transportation systems (see references [1] through [5]). To be more specific, we have attempted to establish a general flexible design philosophy for the control of the positions and the velocities of high-speed automated vehicles during their motion in a single guideway and during merging. Our approach has been to apply available results in the theory of optimal control to this class of problems. Under non-emergency operating conditions, the string of vehicles can be modelled by a set of linear, time-invariant, differential equations. Furthermore, it is possible to incorporate safety and efficiency requirements in a quadratic control cost functional. In this manner, one can readily use the known results on the optimal control of linear systems with respect to quadratic criteria.

This paper represents an effort toward establishing an orderly approach to the problem of merging two or more strings of high-speed vehicles into a single guideway or lane. The position and velocity control of vehicles in a single guideway has been considered before in Refs. [1], [2], and [3]. The equations of the string as well as the performance index is identical to those used by Levine and Athans [1]. For this reason, no extensive motivation and discussion of the criterion is presented.

* This research was conducted at the MIT Electronic Systems Laboratory with support extended by the U.S. Department of Transportation under Contract C-85-65.

In both the merging and the single guideway problem a control cost functional is formulated based on the following broad specifications :

1. Each pair of physically adjacent vehicles must be separated by a given desired separation distance Δ ; any (positive or negative) deviation from Δ should be penalized.
2. Each vehicle must travel with a given string velocity $\bar{v}$; any (positive or negative) velocity deviation from $\bar{v}$ should be penalized.
3. Severe accelerating or decelerating forces should be avoided except in emergency situations.

It is emphasized that emergency control is not considered here.

2. THE MERGING PROBLEM

The physical aspects of the merging problem are illustrated in Figure 1.

1. Two strings of vehicles moving (from right to left) in two single-lane

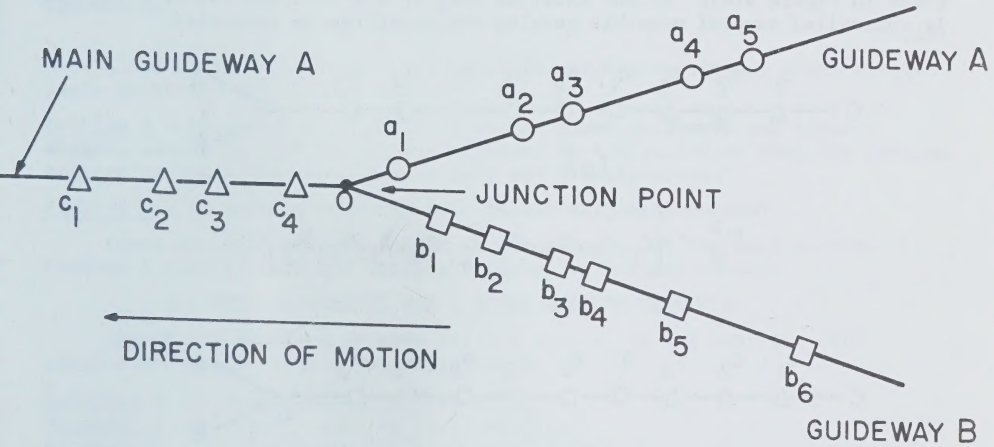


Figure 1 : Visualization of the merging problem and the indexing of the vehicles

guideways A and B must merge into a single-lane guideway C. The junction point is denoted by O. It is assumed that a command and control center is located near the junction so as to automatically control a finite number of vehicles in the guideways A, B, and C which, during some interval of time, are in the vicinity of the junction point.

The vehicles involved in the merging are denoted as follows :

Guideway A : $a_1, a_2, \dots, a_\alpha$

Guideway B : $b_1, b_2, \dots, b_\beta$

Guideway C : $c_1, c_2, \dots, c_\gamma$

For notational purposes we denote by A the set of vehicles in the guideway A. Thus,

$$A = \{a_1, a_2, \dots, a_\alpha\} \quad (2.1)$$

Similarly define

$$B = \{b_1, b_2, \dots, b_\beta\} \quad ; \quad C = \{c_1, c_2, \dots, c_\gamma\} \quad (2.2)$$

It is evident from Figure 1 that distances from the junction point O

can be used to define a natural ordering of the sets A,B,C. The natural ordering is defined by the relations

$$a_1 > a_2 > \dots > a ; \quad b_1 > b_2 > \dots > b ; \quad c_1 > c_2 > \dots > c \quad (2.3)$$

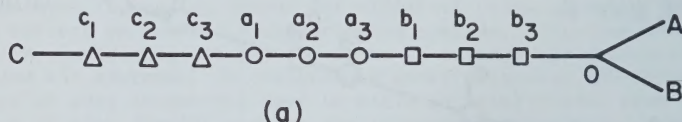
and so one can define the ordered sets A^0 , B^0 , C^0 by

$$A^0 = \{a_i \in A : a_i > a_{i+1} ; i = 1, 2, \dots, \alpha-1\} \quad (2.4)$$

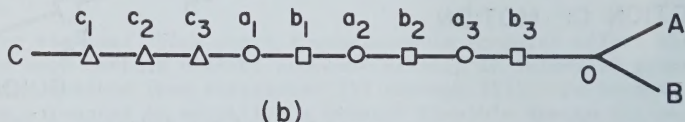
$$B^0 = \{b_i \in B : b_i > b_{i+1} ; i = 1, 2, \dots, \beta-1\} \quad (2.5)$$

$$C^0 = \{c_i \in C : c_i > c_{i+1} ; i = 1, 2, \dots, \gamma-1\} \quad (2.6)$$

If one has absolute control over the vehicles, then one can merge them in many ways. For example, one can move through the junction first all of the vehicles in A and then all the vehicles in B; in this case, after all the vehicles have entered the main guideway, the order depicted in Figure 2(a) will exist. On the other hand, one may wish to interlace the vehicles as shown in Figure 2(b). At any rate, as long as the vehicles can be effectively controlled several possible merging sequences can be generated.



$$P_a^0 = \{c_1, c_2, c_3, a_1, a_2, a_3, b_1, b_2, b_3\}$$



$$P_b^0 = \{c_1, c_2, c_3, a_1, b_1, a_2, b_2, a_3, b_3\}$$

Figure 2 : Two distinct possible merging sequences of six vehicles, initially equally distributed in guideways A and B and their associated ordered sets. In this case, there are 20 possible merging sequences.

The fact that there are many ways of merging the vehicles in A and B can be described mathematically in the following way. Let N denote the total number of vehicles involved, i.e.,

$$N \triangleq \alpha + \beta + \gamma \quad (2.7)$$

Let Q be the unordered set formed by the $(\alpha+\beta)$ vehicles in A and B, i.e.,

$$Q \triangleq \{q_1, q_2, \dots, q_{\alpha+\beta}\} = A \cup B \quad (2.8)$$

Define the ordering : $q_1 > q_2 > \dots > q_{\alpha+\beta}$ (2.9)

and the ordered set Q^0

$$Q^0 = (A \cup B)^0 = \{q_j \in Q : q_j > q_{j+1} ; j = 1, 2, \dots, \alpha+\beta-1\} \quad (2.10)$$

Furthermore, define the ordered sets

$$Q_i^0 = \{a_i \in A, b_i \in B : (A \cup B)^0 - B = A^0 \text{ and } (A \cup B)^0 - A = B^0\} \quad (2.11)$$

where $i = 1, 2, \dots, M$. There in turn can be used to define the ordered sets

$$P_i^0 = \{C^0, Q_i^0\} \quad (2.12)$$

It should be clear (see also Figure 2 for the example) that each set P_i^0 has the property that

- (a) it has $N = \alpha + \beta + \gamma$ elements,
- (b) its first γ elements constitute the ordered set C^0 of Eq. (2.6),
- (c) its last $\alpha + \beta$ elements involve elements of A and B such that if all the elements of B are removed, then the elements of A form the ordered set A^0 and if all the elements of A are removed, then the elements of B form the ordered set B^0 .

Note that the total number M of the sets Q_i^0 (or P_i^0) defines the total number of possible merging sequences. It turns out that M is given by the expression

$$M = \frac{(\alpha + \beta)!}{\alpha! \beta!} \quad (2.13)$$

Since there are a total of M possible merging sequences, there are two basic problems that must be solved.

Problem 1 : Given a set of vehicles in the three guideways and given a merging sequence. How should the vehicles be controlled so that the desired merging sequence takes place in a safe and orderly manner?

Problem 2 : Is there a best (in some sense) merging sequence?

These two problems are solved in the sequel. In the next section Problem 1 will be attacked using the theory of optimal control.

3. OPTIMAL CONTROL FOR A GIVEN MERGING SEQUENCE

Denote the position at some initial time t_0 of the vehicles with respect to the junction point as follows

$$\begin{aligned} \text{Guideway A} &: z[a_1](t_0), z[a_2](t_0), \dots, z[a_\alpha](t_0) \\ \text{Guideway B} &: z[b_1](t_0), z[b_2](t_0), \dots, z[b_\beta](t_0) \\ \text{Guideway C} &: z[c_1](t_0), z[c_2](t_0), \dots, z[c_\gamma](t_0) \end{aligned} \quad (3.2)$$

It is assumed that distances along guideway C are positive, while distances along A and B are negative. Thus,

$$\begin{aligned} 0 &> z[a_1](t_0) > z[a_2](t_0) > \dots > z[a_\alpha](t_0) \\ 0 &> z[b_1](t_0) > z[b_2](t_0) > \dots > z[b_\beta](t_0) \\ z[c_1](t_0) &> z[c_2](t_0) > \dots > z[c_\gamma](t_0) > 0 \end{aligned} \quad (3.2)$$

The true velocities of each vehicle at t_0 will be denoted by

$$v[a_1](t_0), v[b_1](t_0), v[c_k](t_0) \quad (3.3)$$

Assume that all the vehicles are identical. Since all vehicles must travel at the same desired velocity $\bar{v}$, then there is a constant force $\bar{f}$ needed to overcome the drag at the velocity. For small deviations of the actual velocity $v(t)$ of any vehicle from the desired velocity $\bar{v}$ one has^{1,2}

$$\frac{d}{dt} \phi(t) = -\beta \phi(t) + u(t) \quad (3.4)$$

$$\text{where } \phi(t) \triangleq v(t) - \bar{v} \quad (3.5)$$

β : drag coefficient at speed $\bar{v}$ ($\beta > 0$)

$u(t)$: incremental acceleration

Next consider any arbitrary merging sequence. As noted in Section 2 any given merging sequence defines which adjacent vehicles in guideway A or B will continue to be adjacent after they pass through the junction point. Thus, any two vehicles which will become adjacent in guideway C must be controlled in such a way so that any errors in their desired separation are to be eliminated once they go through the junction point. It, therefore, makes sense to control these vehicles prior to the junction point, because, given sufficient time, future separation errors can be corrected using smooth acceleration or deceleration forces (which are important from the passenger-comfort and energy-expenditure viewpoints).

The mathematical formulation of the system equations will now be discussed. Given any one merging sequence indexed by i ($i = 1, 2, \dots, M$ where M is given by Eq. (2.13)). This merging sequence defines an ordered set P_i^0 (see Eq. (2.12)). Let p_{k_i} denote the elements of P_i^0

$$P_i^0 = \{p_{1_i}, p_{2_i}, \dots, p_{(\alpha+\beta+\gamma)_i}\} \quad (3.6)$$

This particular order of merging defines which vehicles are to be adjacent after merging has occurred. So let k_i and k_i+1 ($k_i=1, 2, \dots, \alpha+\beta+\gamma$) index adjacent vehicles under this ordering. Clearly,

$$\frac{d}{dt} z[k_i](t) = v[k_i](t) \quad (3.7)$$

$$\frac{d}{dt} z[k_i+1](t) = v[k_i+1](t) \quad (3.8)$$

Let $e_{k_i}(t)$ denote the deviation of the k_i and k_i+1 vehicles from their desired separation, i.e.,

$$e_{k_i}(t) \triangleq z[k_i](t) - z[k_i+1](t) - \Delta \quad (3.9)$$

Then, this position error satisfies the differential equation

$$\frac{d}{dt} e_{k_i}(t) = v[k_i](t) - v[k_i+1](t) \quad (3.10)$$

Define $\phi_{k_i}(t)$ to be the velocity deviation of the k_i -th vehicle from the desired velocity $\bar{v}$ and let $u_{k_i}(t)$ be the acceleration applied to the same vehicle. Thus,

$$\phi_{k_i}(t) \triangleq v[k_i](t) - \bar{v} \quad (3.11)$$

$$\text{Then } v[k_i](t) - v[k_i+1](t) = \phi_{k_i}(t) - \phi_{k_i+1}(t) \quad (3.12)$$

and so Eqs. (3.10) and (3.4) yield

$$\begin{aligned} \frac{d}{dt} \phi_{k_i}(t) &= -\beta \phi_{k_i}(t) + u_{k_i}(t) \\ \frac{d}{dt} e_{k_i}(t) &= \phi_{k_i}(t) - \phi_{k_i+1}(t) \end{aligned} \quad (3.13)$$

For the given i -th merging sequence, define the state vector $\underline{x}_i(t)$ as the vector obtained by interlacing the velocity and position errors. Thus,

$$\underline{x}_i(t) \triangleq [\phi_{1_i}(t) \ e_{1_i}(t) \ \phi_{2_i}(t) \ \dots \ \phi_{(\alpha+\beta+\gamma)_i}(t)]' \quad (3.14)$$

Similarly define the control vector

$$\underline{u}_i(t) = [u_{1_i}(t) \ u_{2_i}(t) \ \dots \ u_{(\alpha+\beta+\gamma)_i}(t)]' \quad (3.15)$$

In this case the state vector $\underline{x}_i(t)$ obeys the vector differential equation

$$\frac{d}{dt} \underline{x}_1(t) = \underline{A} \underline{x}_1(t) + \underline{B} \underline{u}_1(t) \quad (3.16)$$

where $\underline{A}$ is the $[2(\alpha+\beta+\gamma)-1] \times [2(\alpha+\beta+\gamma)-1]$ matrix

$$\underline{A} = \begin{bmatrix} -\beta & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & -\beta & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -\beta \end{bmatrix} \quad (3.17)$$

and $\underline{B}$ is the $[2(\alpha+\beta+\gamma)-1] \times (\alpha+\beta+\gamma)$ matrix

$$\underline{B} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \quad (3.18)$$

It is important to realize that the above formulation reduces the problem of controlling the vehicles in the merging problem to the problem of controlling vehicles in a single string. The only difference is that this single string has the property that vehicles belong to the guideways A and B have the freedom of "moving over each" other as long as they are physically in different guideways prior to merging.

To illustrate this idea consider the situation shown in Figure 3. In Fig. 3(a) one has seven vehicles as they appear at the starting time t_0 . In Fig. 3(b) one sees the implications of a particular merging sequence after merger has occurred. Figure 3(c) shows the vehicles at $t = t_0$ as they appear on a fictitious "single-lane" guideway from the viewpoint of their distances from the merging point 0. It then appears that vehicle a_1 must "jump-over" b_1 , and b_2 must "jump-over" a_2 to obtain the merging sequence demanded by Fig. 3(b). This "jumping-over" phenomenon is allowed as long as it occurs prior to the reaching of the junction point of the vehicles involved.

The requirement of controlling the vehicles so that they merge according to a desired merging sequence while they are properly separated and travel near the desired velocity $\bar{v}$ can be accomplished by defining a cost functional J_1 (depending on the merging sequence) of the following quadratic type.

$$J_1 = \int_{t_0}^{\infty} \{ q \left[\sum_{k_1=1}^{\alpha+\beta+\gamma-1} e_{k_1}^2(t) \right] + p \left[\sum_{k_1=1}^{\alpha+\beta+\gamma} \phi_{k_1}^2(t) \right] + r \left[\sum_{k_1=1}^{\alpha+\beta+\gamma} u_{k_1}(t) \right] \} dt \quad (3.19)$$

$$q > 0, p > 0, r > 0$$

The choice of this type of performance criterion has been discussed extensively previously^{1,2,3} and, for this reason, is not included here. In essence, it is the opinion of the author, that it accurately reflects any relative weighing desired in the broad specifications discussed in Section 1 under non-emergency operating conditions.

The form of the control-cost functional J_1 of Eq. (3.19) leads to the definition of the $[2(\alpha+\beta+\gamma)-1] \times [2(\alpha+\beta+\gamma)-1]$ positive semidefinite matrix

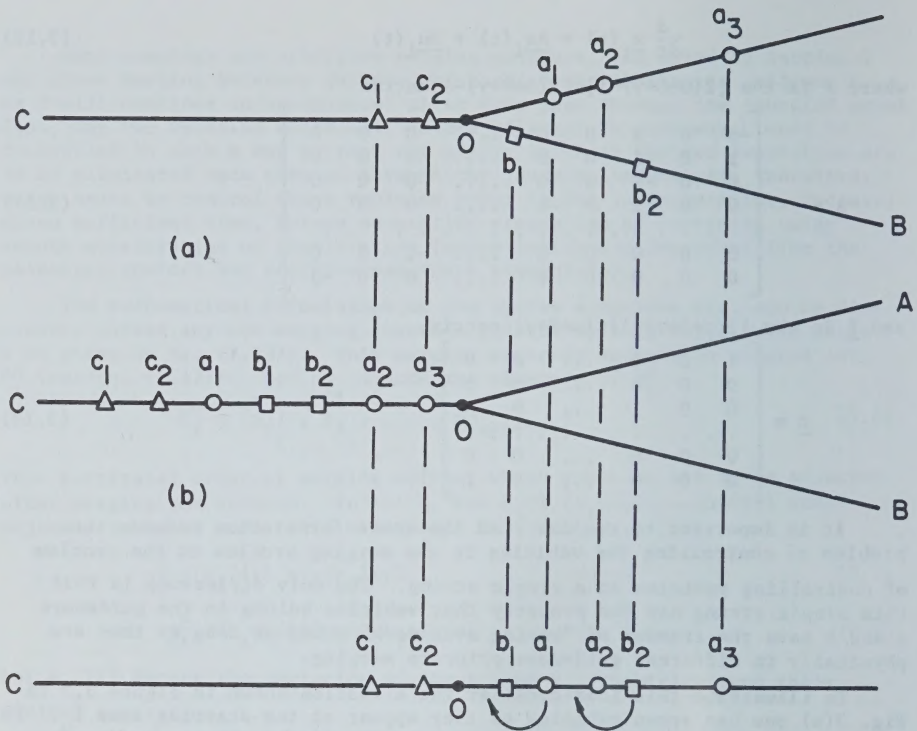


Figure 3 : (a) Initial vehicle distribution; (b) desired merging sequence; (c) equivalent "single-guideway" control problem.

$$\underline{Q} = \begin{bmatrix} p & 0 & 0 & \dots & 0 & 0 \\ 0 & q & 0 & \dots & 0 & 0 \\ 0 & 0 & p & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & p \end{bmatrix} \quad (3.20)$$

and of the $(\alpha+\beta+\gamma) \times (\alpha+\beta+\gamma)$ matrix $\underline{R}$

$$\underline{R} = r \underline{I} \quad (3.21)$$

Using these two definitions the cost functional J_1 of Eq. (3.19) can be written in the form

$$J_1 = \int_{t_0}^{\infty} \{ \underline{x}'_1(t) \underline{Q} \underline{x}_1(t) + \underline{u}'_1(t) \underline{R} \underline{u}_1(t) \} dt \quad (3.22)$$

It is well known⁶ that the control which minimizes the cost functional J_1 is given by (under the above formulation) by

$$\underline{u}_1^0(t) = -\frac{1}{r} \underline{B}' \underline{K} \underline{x}_1(t) \triangleq -\underline{G} \underline{x}_1(t) \quad (3.23)$$

and the minimum value J_1^0 of the cost functional J_1 at $t=t_0$ is given by

$$J_1^0 = \underline{x}'_1(t_0) \underline{K} \underline{x}_1(t_0) \quad (3.24)$$

where $\underline{K}$ is the unique positive definite solution of the matrix equation

$$-\underline{K} \underline{A} - \underline{A}' \underline{K} + \frac{1}{r} \underline{K} \underline{B} \underline{B}' \underline{K} - \underline{Q} = \underline{0} \quad (3.25)$$

The determination of the solution-matrix $\underline{K}$ of the above equation can be

accomplished in a straight forward manner using a digital computer. Thus, given the number of vehicles, their drag coefficient β , and the weighting constants p, q , and r in the cost functional (3.19), one can determine the matrix $\underline{K}$; this in turn defines the gain matrix $\underline{G}$ (see Eq. (3.23))

$$\underline{G} = \frac{\Delta}{r} \underline{B}' \underline{K} \quad (3.26)$$

It is important to emphasize that the matrices $\underline{K}$ and $\underline{G}$ are independent of the merging sequence, i.e. that these two matrices are completely independent of the index i which distinguishes one merging sequence from another.

The only effect that a particular merging sequence has is in the construction of the state vector $\underline{x}_i$ and of the control vector $\underline{u}_i$. The pre-computation of the gain matrix $\underline{G}$ and the definition of the state vector $\underline{x}_i$ for a particular merging sequence immediately define the best possible control $\underline{u}_i^0(t) = -\underline{G}\underline{x}_i(t)$ for that particular merging sequence. From a physical point of view, this means that in two distinct merging sequences, the measured state variables (position and velocity errors) are weighted by different combinations of the same feedback gains to produce the appropriate control accelerations or decelerations.

This completes the solution of the first posed problem, namely, how to control for any given merging sequence. The question of determining the best possible merging sequence is treated in the following section.

4. DETERMINATION OF THE BEST MERGING SEQUENCE

The fact that one can use quadratic performance criteria for controlling the vehicles for any given sequence implies that this method of control is satisfactory with respect to the broad specifications stated in section 1. Indeed, the simulation results reported in Refs. [1], [2] and [3] reinforce the validity of this type of criterion from the transient response point of view. Therefore, it is reasonable to use the same criterion to determine the best possible merging sequence.

As before, let i denote a particular merging sequence. Let $\underline{x}_i(t_0)$ denote the state vector induced by this sequence. If one uses the optimal control scheme described above, then (see Eq. (3.24)) the cost

$$J_i^0 = \underline{x}_i'(t_0) \underline{K} \underline{x}_i(t_0) \quad (4.1)$$

is the minimum possible control cost associated with the i -th merging sequence. Use of this scalar control cost can thus be used to decide which of two merging sequences is "better." Thus, consider another merging sequence with associated state vector $\underline{x}_k(t_0)$, with $k \neq i$; the minimum control cost associated with this sequence is

$$J_k^0 = \underline{x}_k'(t_0) \underline{K} \underline{x}_k(t_0) \quad (4.2)$$

$$\text{If } J_k^0 < J_i^0 \quad (4.3)$$

then one can deduce that the k -th merging sequence is superior to the i -th merging sequence, because the latter will have more errors and control efforts than the former.

Once one has accepted the ordering of the merging sequences on the basis of their associated minimum control costs, then it becomes a trivial matter to define the optimal merging sequence. This can be done as follows: suppose that a merging sequence indexed by i^* has the property that

$$J_{i^*}^0 < J_i^0 \quad \text{for all } i = 1, 2, \dots, M \quad (4.4)$$

Then i^* indexes the optimal merging sequence.

To determine the optimal merging sequence one must merely compute the

M scalars $J_1^0, J_2^0, \dots, J_M^0$ by

$$\begin{aligned} J_1^0 &= \underline{x}_1'(t_0) \underline{K} \underline{x}_1(t_0) \\ J_2^0 &= \underline{x}_2'(t_0) \underline{K} \underline{x}_2(t_0) \\ &\dots \dots \dots \\ J_M^0 &= \underline{x}_M'(t_0) \underline{K} \underline{x}_M(t_0) \end{aligned} \quad (4.5)$$

and select the smallest one. This procedure defines the optimal merging sequence indexed by i^* , its corresponding state vector $\underline{x}_{i^*}$, and the associated optimal feedback control scheme given by

$$\underline{u}_{i^*}(t) = -\underline{R}^{-1} \underline{B}' \underline{K} \underline{x}_{i^*}(t) = -\underline{G} \underline{x}_{i^*}(t) \quad (4.6)$$

In this manner, one has a systematic computerized solution to the selection of the best possible control sequence and of the control system which will insure that it takes place in a smooth and efficient manner.

5. DISCUSSION

The solution to the posed problem involves a certain amount of on-line computation that must be performed at the command and control center. The computational aspects of the problem will be discussed below.

One of the major simplifying factors inherent in this problem is that the $\underline{K}$ matrix and the gain matrix $\underline{G}$ are independent of the actual merging sequence. Thus, knowledge of (a) the total number of trains $N = \alpha + \beta + \gamma$ in the three guideways, (b) the mass m and drag coefficient β of each train, and (c) the weighting factors p, q , and r in the cost functional (3.19) is sufficient to generate $\underline{K}$ and $\underline{G}$. Thus, these two matrices can be precomputed and stored in the computer memory.

The determination of the best merging sequence, based on the comparison of the control costs (4.5), requires the formation of the state vectors $\underline{x}_i(t_0)$, $i = 1, 2, \dots, M$ where $M = (\alpha + \beta)! / \alpha! \beta! \dots$. In essence, each merging sequence assigns an index to each vehicle and defines which vehicles will be adjacent after the merging has been completed. The state vector $\underline{x}_i(t_0)$ is then formed by first measuring the position and velocity errors using Eqs. (3.9) and (3.10) and then interlacing them as indicated by Eq. (3.14). Thus the formation of the state vector for each merging sequence requires a total of $2N-1$ additions. The determination of each control cost J_i^0 requires a total of $(2N-1)^2 + 2N-1$ multiplications and $(2N-1)^2 + 2N-2$ additions. Thus to evaluate the control cost for each merging sequence one needs to perform $4N^2 - 2$ additions and $4N^2 - 2N$ multiplications. Since there are M possible merging sequences then it follows that the maximum number of operations required to determine the best merging sequence is $M(4N^2 - 2)$ additions and $M(4N^2 - 2N)$ multiplications. As an example, suppose that one deals with a total of 12 vehicles ($N=12$) five in guideway A, five in guideway B ($\alpha=\beta=5$) and two in the main guideway C. Then, the total number of the possible merging sequences is $M = 252$ and so to determine the best merging sequence one needs 144,648 additions and 139,104 multiplications. Assuming* that each floating point addition operation requires, on the average, 10 microseconds and each floating point multiplication requires 5 microseconds, it is easy to deduce that in this example one can determine the best merging sequence in less than 3 seconds of computer time by comparing all possible 252 merging sequences. Of course, any setting of priorities and/or additional statistical information regarding the position and velocity error

* These figures are typical for the IBM 7094 digital computer.

distribution can be used to drastically reduce the merging sequences that should be considered and, hence, the required computation time. It therefore appears that the computational requirements for optimal merging are not unrealistic with respect to current computer technology.

6. CONCLUSIONS

An orderly and systematic procedure has been proposed for the control of high-speed vehicles during the merging phase through the use of optimal control theory. It has been demonstrated that the computations required to determine the best merging sequence are not unrealistic.

ACKNOWLEDGMENT

The author wishes to express his gratitude to Dr. A.H. Levis and to Mr. W.S. Levine for their many contributions to this project. Thanks are also due to Mr. L.S. Kramer and Prof. R.N. Spann for their comments on this problem.

REFERENCES

1. Levine, W.S. and Athans, M., "On the Optimal Error Regulation of a String of Moving Vehicles," IEEE Trans. on Automatic Control, Vol. AC-11, No. 3, July 1966, pp. 355-361.
2. Athans, M., Levine, W.S., and Levis, A.H., "On the Optimal and Suboptimal Position and Velocity Control of a String of High-Speed Vehicles," Report ESL-R-291, M.I.T. Electronic Systems Lab., Cambridge, Mass., November, 1966.
3. Athans, M., Levine, W.S., and Levis, A.H., "A System for the Optimal and Suboptimal Position and Velocity Control for a String of High-Speed Vehicles," Proc. Fifth International Congress of AICA, Lausanne, Switzerland, August 1967, (to be published).
4. Athans, M. and Levine, W.S., "Optimal Vehicle Control for the Merging Problem," Report ESL-R-327, M.I.T. Electronic Systems Laboratory, Cambridge, Mass., 1967.
5. Levis, A.H. and Athans, M., "Sampled Data Control of High Speed Trains," Report ESL-R-339, M.I.T. Electronic Systems Lab Report, Cambridge, Mass., 1968.
6. Athans, M. and Falb, P.L., Optimal Control, McGraw-Hill Book Co., Inc. New York, N.Y., 1966.

AN OPTIMAL CONTROL ALGORITHM FOR RAMP METERING OF URBAN FREEWAYS

LI SHIN YUAN AND JOHN B. KREER
Michigan State University
East Lansing, Michigan

ABSTRACT

An urban freeway is treated as a dynamic process. A state model for the freeway is obtained with sectional traffic densities as states and entrance flow rates as controls. A linear programming problem is solved to obtain the optimal freeway densities and entrance flow rates under steady-state conditions, and a state regulator is used to minimize the deviations in traffic densities from these optimal steady state values.

INTRODUCTION

It is well established by theoretical and experimental work that a plot of the steady-state flow rate, y , (in vehicles per hour) as a function of the traffic density, x , (in vehicles per mile) for a long uniform section of freeway or street with no exit or input points is of the general form shown in Figure 1.¹ It follows that the density on any freeway will eventually exceed the point of maximum flow rate if access is uncontrolled under heavy demand. Consequently, maximum effective use of any freeway facility, or surface street network, whether of good or bad design and construction, can be realized only by controlling its loading. With the high cost of construction of urban freeways, even a modest increase in the efficiency of their operation will result in a considerable economic gain.

A method called ramp metering^{2, 3} is being used in some cities to keep the freeway density below the critical value by controlling entrance ramp flow rates. However, existing ramp metering systems base their control action only on conditions in the immediate vicinity of the individual entrance ramps. For an alternative this paper describes a freeway control algorithm which coordinates the entrance ramp flow rates used by the individual controllers to:

- (1) maximize the number of vehicles served under conditions of over-demand.
- (2) balance the lengths of the ramp queues created by access control of the freeway.
- (3) suppress the effects of random disturbances in the traffic flow.

Such a system would be implemented by placing vehicle presence detectors at appropriately selected locations along the freeway to sense vehicle density. This information would be transmitted to a digital processor which would set the ramp metering rates according to the algorithm to be described.

DEVELOPMENT OF A STATE MODEL OF A FREEWAY

The dynamic behavior of a freeway as required for on-line control is dominated by two mechanisms: (i) conservation of vehicles, and (ii) drivers in the traffic stream reacting to increasing density by reducing their speed. Mechanism (i) can be expressed mathematically by

$$\frac{\partial x}{\partial t} + \frac{\partial y}{\partial z} = 0 \quad (1)$$

where z is the position coordinate along the freeway. Equation (1) can be interpreted as saying that for a short section of freeway the rate of change of flow with respect to position along the freeway, $\partial y/\partial z$, is proportional to the difference between the flows into and out of the section. If this difference is nonzero, a change in density in the section as a function of time will be observed.

At any given point the flow and density are related by

$$y = vx \quad (2)$$

where v is the speed of the vehicles. A number of relationships have been proposed for quantitatively describing (ii). Based on the data taken from the Lodge Freeway¹ the linear model of Greenshields appears to be the most realistic for these purposes. Thus the speed will be given by

$$v = v_f \left(1 - \frac{x}{x_j}\right) \quad (3)$$

where v_f is the free speed, the limiting value of speed as density approaches zero. The jam density, at which all vehicles will come to a halt, is denoted by x_j . Typically x_j is approximately 40% of the bumper to bumper density.

Equation (2) is a per lane relationship. If at same z , the lane densities are assumed to be uniform, (2) combined with (3) can be modified so that

$$y = \ell(z) v_f \left(1 - \frac{x}{x_j}\right) x, \quad \text{for } x \leq x_j \quad (4)$$

where y and $\ell(z)$ are the total flow rate and the number of lanes in each direction at z .

Entrance and exit ramps are assumed to cause a discontinuity in the freeway stream flow by the amount of the ramp flow. The effects of the ramps will enter the model by the presence in $\partial y/\partial z$ of terms of the form

$$- \sum_j y_i^j \delta(z^j) + \sum_k y_o^k \delta(z^k)$$

where $\delta(z^j)$ is the Dirac impulse function, z^j is the location of the j th entrance ramp, and y_i^j and y_o^k are the flows of input ramp j and output ramp k respectively.

DISCRETIZATION OF THE FREEWAY MODEL

As indicated by (1) and (4) a freeway is a nonlinear, distributed parameter system. The analysis which follows will employ spatial discretization of (1). Instead of discretizing into sections of uniform length, the boundaries between sections of the freeway are assumed to be chosen so that all exit ramps, entrance ramps, and changes in number of lanes occur at the boundaries of sections. Additional section boundaries may be added at points of pronounced change in geometric features of the freeway which could be expected to affect the flow of traffic. With these assumptions as to the location of the section boundaries,

for each section (1) can be approximated by

$$\frac{dx^k}{dt} = \frac{1}{d^k} [y^{k-1} - \delta_o^{k-1} y_o^{k-1} - y^k + \delta_i^k y_i^k] \quad (5)$$

where

y^k = the flow rate at the downstream boundary of section k

x^k = the density in section k corresponding to y^k

y_i^k = the input rate of the entrance ramp of section k

y_o^k = the flow rate of the output ramp of section k

$\delta_i^k = 1$ if an entrance ramp is located in section k,
 0 if no entrance ramp is located in section k

$\delta_o^k = 1$ if an exit ramp is located in section k,
 0 if no exit ramp is located in section k

d^k = the physical length of section k

y^o = flow rate from uncontrolled section of the freeway into section 1

$k = 1, 2, \dots, n$

Since the entrance and exit ramps always occur at a section boundary, the convention has been adopted to assign them to sections such that an entrance (exit) ramp always is at the upstream (downstream) end of the section. y^n is the outflow at the end of the freeway onto the city street system or outbound into the intercity portion of the freeway network. Assuming that, at any exit ramp, a known fraction $f^k(t)$, of the total flow will leave the freeway, the exit flow at ramp k is

$$y_o^k = f^k y^k \quad (6)$$

Using (4) and (6) in (5), it becomes

$$\frac{dx^1}{dt} = \frac{1}{d^1} [y^o - \ell^1 v_f^1 (x^1 - \frac{[x^1]^2}{x_j^1}) + \delta_i^1 y_i^1] \quad (7a)$$

$$\begin{aligned} \frac{dx^k}{dt} = \frac{1}{d^k} [\ell^{k-1} v_f^{k-1} (x^{k-1} - \frac{[x^{k-1}]^2}{x_j^{k-1}}) (1 - \delta_o^{k-1} f^{k-1}) \\ - \ell^k v_f^k (x^k - \frac{[x^k]^2}{x_j^k}) + \delta_i^k y_i^k] \end{aligned} \quad (7b)$$

$k = 2, 3, \dots, n$

where y^o is the inflow from the uncontrolled portion of the freeway at the upstream end.

In compact form:

$$\frac{dx}{dt} = F(x) + B y_i \quad (8)$$

where

$F(x)$ is a vector whose components are the terms on the right hand side of (7) involving x^k

$$B = \text{diag} \left[\frac{\delta_i^1}{d^1}, \frac{\delta_i^2}{d^2}, \dots, \frac{\delta_i^n}{d^n} \right].$$

In (8), the symbol $\delta_i^1 y_i^1$ is understood to include the contributions of both y^0 and $\delta_i^1 y_i^1$ in (7a).

OPTIMAL CONTROL OF THE FREEWAY

It is reasonable to assume that after the freeway is under controlled, the entrance flow rates as well as the traffic densities along the freeway will finally approach some steady state values. Therefore, the control vector $y_i(t)$ is divided into a steady state component, called a reference component, which is constant during each control interval and a time varying component. The reference component is selected on the basis of maximizing the total number of vehicles served, and also balancing the entrance queues using a steady state model of the system which considers the nonlinearities.

For this reference value of control vector there will be a corresponding steady state density in each section of the freeway. A local linearization of the freeway model is performed about this reference value of density. The varying component of the control vector is determined using standard linear regulator techniques based on the linearized model and a quadratic performance functional of deviations from reference density and reference ramp flow rates.

The philosophy used here is first to find an optimal steady state density vector and then regulate the entrance ramp rates to keep the state of the system near this vector.

The steady state model can be derived by setting dx^k/dt to zero in (5), eliminating $\delta_o^k y_o^k$ using (6), and setting $y_i^k = y_{ir}^k$. The result is

$$y_r^1 = \delta_i^1 y_{ir}^1 \quad (9a)$$

$$y_r^k = (1 - \delta_o^{k-1} f^{k-1}) y_r^{k-1} + \delta_i^k y_{ir}^k \quad (9b)$$

The dependent variables which are subject to control are y_r^k for $k = 1, 2, \dots, n$. By recursive substitution, explicit expressions for y_r^k are obtained as:

$$\begin{aligned} y_r^k = & (1 - \delta_o^{k-1} f^{k-1})(1 - \delta_o^{k-2} f^{k-2}) \dots (1 - \delta_o^1 f^1) \delta_i^1 y_{ir}^1 \\ & + (1 - \delta_o^{k-1} f^{k-1})(1 - \delta_o^{k-2} f^{k-2}) \dots (1 - \delta_o^2 f^2) \delta_i^2 y_{ir}^2 \\ & + \dots + \delta_i^k y_{ir}^k, \end{aligned} \quad (10)$$

$$k = 2, 3, \dots, n$$

The optimal values for y_{ir} are obtained by solving the following linear programming problem:

Maximize $\sum_{k=1}^n (c_q^k q_i^k + c_d^k d_i^k) y_{ir}^k$ subject to the constraints

$$y_r^k \leq y_m^k, \quad k = 1, 2, \dots, n. \quad (11)$$

$$0 \leq y_{ir}^k \leq y_{im}^k, \quad \text{for all } k \text{ such that } q_i^k > 0. \quad (12)$$

$$0 \leq y_{ir}^k \leq \min(y_{im}^k, d_i^k) \text{ for all } k \text{ such that } q_i^k = 0. \quad (13)$$

where q_i^k and d_i^k are the queue length and demand rate respectively at entrance ramp k , c_q and c_d are two constant weighting vectors, y_m and y_{im} are the maximum allowable flow rate and entrance flow rate respectively and y_r^k are expressed in terms of y_{ir} by (9a) and (10).

Note that the object function of the linear programming problem always gives higher priorities to entrance ramps with longer queues and higher demands, so it is designed to balance the queues at the entrance ramps while maximizing the freeway service.

Once y_{ir} has been determined by linear programming, the corresponding value for x_r can be found by using (9a) and (10) to find y_r . The reference density for each section can be calculated from the flow-density characteristic, using the lower of the two densities which is possible for the y_r^k , i.e.

$$x_r^k = \frac{1}{2} \left[x_j^k - \sqrt{(x_j^k)^2 - \frac{4x_j^k y_r^k}{f^k v_f^k}} \right] \quad (14)$$

In the absence of any random disturbances in the traffic flow on the freeway maintaining the ramp metering rates at y_{ir} should keep the density at x_r . Since random accelerations and decelerations of vehicles in the traffic stream are certain to occur, it is necessary to superimpose a variable component on the ramp metering rates to regulate the densities to x_r . For this purpose let

$$x(t) = x_r + e(t) \quad (15)$$

and

$$y_i(t) = y_{ir} + w(t) \quad (16)$$

where $e(t)$ and $w(t)$ are perturbation vectors. Substituting (15) and (16) into (8), expanding $F(x_r + e)$ around x_r by Taylor series expansion, and neglecting the second order terms in e (F is quadratic in x so that the partial derivatives of F with respect to x of order higher than two are zero), one has:

$$\begin{aligned} \dot{e}(t) &= A e(t) + B w(t) \\ e(0) &= x(0) - x_r \end{aligned} \quad (17)$$

where

$$A = \begin{bmatrix} \frac{\partial F^1}{\partial x^1} & \frac{\partial F^1}{\partial x^2} & \frac{\partial F^1}{\partial x^n} \\ \frac{\partial F^n}{\partial x^1} & \frac{\partial F^n}{\partial x^2} & \frac{\partial F^n}{\partial x^n} \end{bmatrix}_{x=x_r} = [a^{ij}] \quad (18)$$

with

$$a^{kk} = \frac{-\ell^k v_f^k}{d^k} \left[1 - \frac{2x_r^k}{x_j^k} \right] \quad (19a)$$

$$a^{k, k-1} = \frac{\ell^{k-1} v_f^k}{d^k} \left[1 - \delta_o^{k-1} f^{k-1} \right] \left[1 - \frac{2x_r^{k-1}}{x_j^{k-1}} \right] \quad (19b)$$

$$a^{k\ell} = 0, \text{ for } \ell \neq k, k-1 \quad (19c)$$

Fortunately, system (17) is completely controllable.⁴ Due to this complete controllability it is well known that an optimal control which minimizes the performance functional

$$J(w) = \frac{1}{2} \int_0^\infty [< e(t), Qe(t) > + < w(t), Rw(t) >] dt \quad (20)$$

(Q is a positive semidefinite and R is a positive definite matrix) exists, is unique, and is given by the equation:

$$w^*(t) = -R^{-1} B^T \hat{K} e^*(t) \quad (21)$$

where $\hat{K}$ is the constant $n \times n$ positive definite matrix which is the solution of the algebraic Riccati equation:

$$-\hat{K}A - A^T \hat{K} + \hat{K}BR^{-1}B^T \hat{K} - Q = 0 \quad (22)$$

the * denotes the optimality and T means tranpose. The optimal entrance flow rate y_i^* is obtained by:

$$y_i^*(t) = y_{ir} + w^*(t) = y_{ir} - R^{-1} B^T \hat{K} e^*(t) \quad (23)$$

The vector $e^*(t)$ is obtained by:

$$e^*(t) = x(t) - x_r \quad (24)$$

where the vector $x(t)$ is measured by the density detectors along the free-way.

IMPLEMENTATION OF THE CONTROL ALGORITHM

The data which the central controller must have in order to compute the optimal entrance ramp rates determined by this algorithm are x , d_i , q_i , x_i , v_f and f . These quantities can all be measured using suitably placed vehicle presence detectors. Densities are determined by accumulating the fraction of time a vehicle is indicated as present by the

detector and multiplying this by the density of average length vehicles which would exist at bumper to bumper density. Demand is measured by counting vehicles passing a detector at a point far enough upstream on the entrance ramp that the queue will not reach it. The queues are measured from the difference between the counts of the demand measuring detectors and the counts of detectors placed at the downstream ends of the ramps. The fraction of the traffic stream leaving at each exit ramp f^k is measured by suitably placed detectors. Speeds are measured by dividing the time a vehicle presence is detected by average vehicle length. By curve fitting to a number of pairs of density and average vehicle speed measurements it is possible to determine values of x_j and v_f . These latter quantities change slowly so that the fact that it takes longer to arrive at an individual measurement of them is not serious.

The flow chart for the computer program which would implement the control algorithm is shown in Figure 2. The vectors x , x_j , v_f , d_i , f and q_i are monitored continuously. Whenever a density above a level judged to be critical is detected in any section of the freeway, the optimal ramp rates are computed and ramp metering is activated. New values of x , d_i , and q_i can be obtained approximately once per minute from the vehicle detectors. The ramp metering is continued until all queues are reduced to zero. New values of x_r and K are computed and used by the controller any time the measurements indicate a significant change in x_j or v_f and when a significant change in the relative demands or queue lengths occurs. The latter is considered to have occurred whenever the inequality

$$\left| \frac{c_q^k q_i^k (m+1) + c_d^k d_i^k (m+1)}{\sum_k [c_q^k q_i^k (m+1) + c_d^k d_i^k (m+1)]} - \frac{c_q^k(m) q_i^k(m) + c_d^k d_i^k(m)}{\sum_k [c_q^k q_i^k(m) + c_d^k d_i^k(m)]} \right| > \epsilon_3$$

m = index of the q_i and d_i measurements

ϵ_3 = suitably chosen threshold value

is satisfied for some value of k .

CONCLUSION

It has been demonstrated that well established techniques of optimal control can be applied to the optimization of ramp metering of urban freeways.

ACKNOWLEDGMENT

The research described in this paper was partially supported by the Crouse-Hinds Company under contract with Michigan State University.

REFERENCES

1. Kreer, J. B. and Goodnuff, J. L., "On the Automatic Control of Freeway Density," IEEE Automotive Conference, Detroit, Michigan, September 1967.
2. May, A. D., "Improving Network Operations with Freeway Ramp Control," 43rd Annual Meeting of the Highway Research Board, Washington, D. C., January 13-17, 1964.
3. McCasland, W. R., Drew, D. R., and Wattleworth, J. A., "Houston Freeway Surveillance and Control Project: 1966 Progress Report." Program Review Meeting, Research and Development of Traffic Systems, Gaithersburg, Md., December 6-8, 1966.
4. Kalman, R. E., "Contributions to the Theory of Optimal Control," Bol. Soc. Mat. Mex., vol. 5, pp. 102-199, 1960.

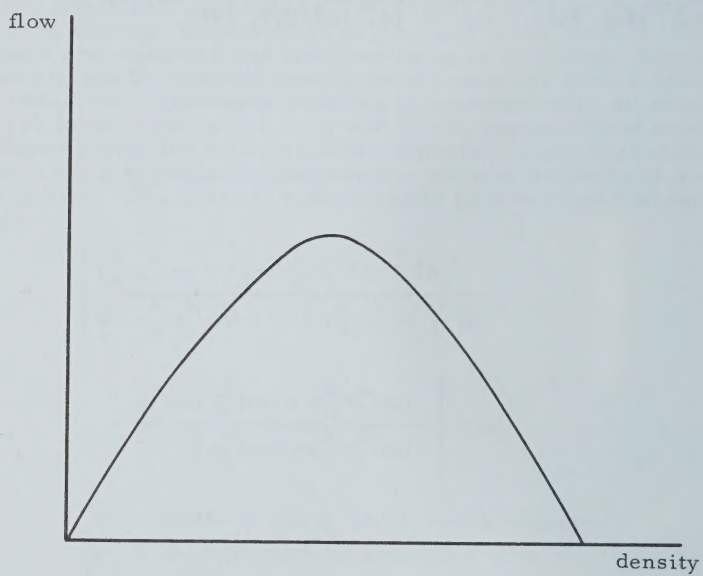


Figure 1. Steady State Flow Density Plot.

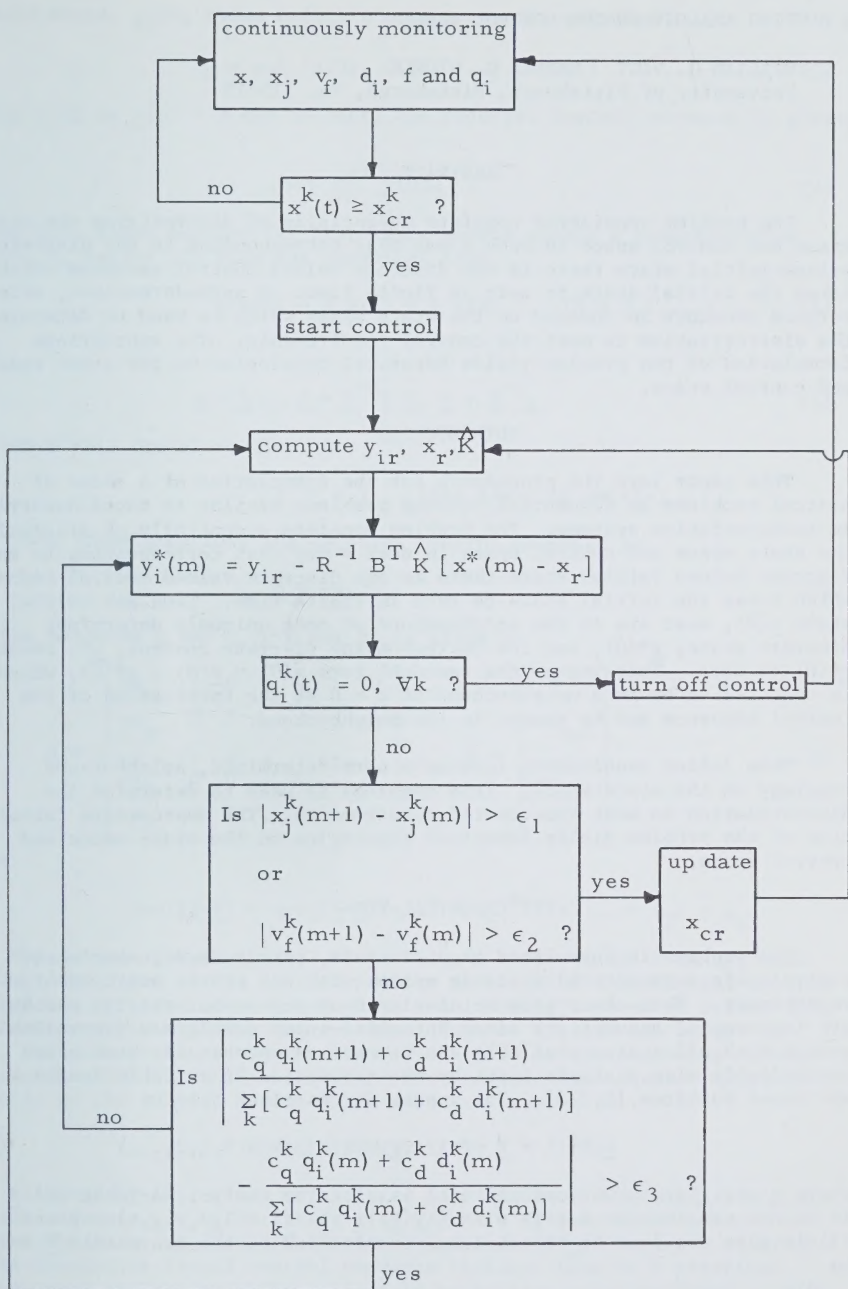


Figure 2. Flow Chart of Optimal Control Algorithm.

A ROUTING ANALOGY IN THE CONTROL SPACE

WILLIAM G. VOGT MARLIN H. MICKLE
University of Pittsburgh, Pittsburgh, Pa. 15213

ABSTRACT

The problem considered consists essentially of discretizing the state space and control space in such a way that corresponding to any discrete valued initial state there is one discrete valued control sequence which takes the initial state to zero in finite time. A norm-determined, neighborhood topology is induced on the state space which is used to determine the discretization to meet the control requirements. The appropriate formulation of the problem yields identical topologies on the state space and control space.

INTRODUCTION

This paper lays the groundwork for the formulation of a class of control problems as sequential routing problems similar to those occurring in transportation systems. The problem consists essentially of discretizing the state space and control space in such a way that corresponding to any discrete valued initial state there is one discrete valued control sequence which takes the initial state to zero in finite time. Thus any initial state $\underline{x}(0)$, must lie in the neighborhood of some uniquely determined discrete state, $\underline{x}^*(0)$, and the corresponding discrete control, $\underline{u}^*$, reduces $\underline{x}^*(0)$ to zero. This leaves the residual term $\underline{\tilde{x}}(0) = \underline{x}(0) - \underline{x}^*(0)$, which is required to be in a neighborhood of $\underline{x} = 0$ at the termination of the control sequence and to remain in the neighborhood.

This latter requirement induces a norm-determined, neighborhood topology on the state space. This topology is used to determine the discretization to meet the control requirements. The appropriate formulation of the problem yields identical topologies on the state space and control space.

THE CANONICAL FORM

The problem is formulated as a discrete, single-input, completely controllable asymptotically stable system with all states available for measurement. That the system be single-input and asymptotically stable are inessential assumptions since any multi-input completely controllable system with all states available can be made asymptotically stable and controllable with a single input by the connection of suitable feedback and input matrices [1,2,3]. The system is described then by

$$\underline{y}(k+1) = \underline{F} \underline{y}(k) + \underline{g} u(k) \quad (k = 0, 1, 2, \dots) \quad (1)$$

where $\underline{y}$ and $\underline{g}$ are n -vectors and $u(k)$ is a scalar control variable and $\underline{F}$ is an $n \times n$ nonsingular matrix with $|\lambda_i(\underline{F})| < 1$ ($i = 1, 2, \dots, n$). Since (1) is also required to be completely controllable, the $n \times n$ matrix $\underline{P}$ defined by

$$\underline{P} = [\underline{F}^{-1} \underline{g}, \underline{F}^{-2} \underline{g}, \dots, \underline{F}^{-n} \underline{g}] \quad (2)$$

is nonsingular. It is well known that under these conditions, given any

initial state, $\underline{y}(0)$, there exists a unique control sequence

$$\underline{u} = \text{col} \{u(0), u(1), \dots, u(n-1)\} \quad (3)$$

taking $\underline{y}(0)$ to $\underline{y}(n) = 0$ and in fact, the required control sequence is given by

$$\underline{u} = -\underline{P}^{-1} \underline{y}(0). \quad (4)$$

It has been shown earlier [4] that there always exists a transformation matrix, $\underline{T}$, taking (1) into the canonical form

$$\underline{x}(k+1) = \underline{A} \underline{x}(k) + \underline{b} u(k) \quad (k = 0, 1, \dots) \quad (5)$$

with

$$\underline{y} = \underline{T} \underline{x}, \underline{A} = \underline{T}^{-1} \underline{F} \underline{T}, \underline{b} = \underline{T}^{-1} \underline{g} \quad (6)$$

and under this transformation

$$\underline{T}^{-1} \underline{P} = \underline{I} \quad (\text{the identity matrix}).$$

With this transformation, the required control sequence is

$$\underline{u} = -\underline{x}(0) \quad (7)$$

and the matrices $\underline{A}$ and $\underline{b}$ turn out to be given by

$$\underline{A} = \begin{bmatrix} -a_n & 1 & 0 & \dots & \dots & 0 \\ -a_{n-1} & 0 & 1 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -a_2 & 0 & 0 & \dots & \dots & 1 \\ -a_1 & 0 & 0 & \dots & \dots & 0 \end{bmatrix}; \quad \underline{b} = \begin{bmatrix} -a_n \\ -a_{n-1} \\ \dots \\ -a_2 \\ -a_1 \end{bmatrix} \quad (8)$$

where

$$\det[\lambda \underline{I} - \underline{F}] = \det[\lambda \underline{I} - \underline{A}] = \lambda^n + a_n \lambda^{n-1} + \dots + a_2 \lambda + a_1$$

The canonical form of (6,7,8) provides a basis on which an approach to the formulation of the control problem in terms of a sequential routing problem can be made. Since, under the requirements imposed on system (1), the transformations giving the canonical form of (6,7,8) can always be made, this canonical representation will be used in the following development. Note that (7) implies that the required control sequence is given directly by the initial state of the canonical system (6).

PROBLEM FORMULATION

Although by (7), given any initial state $\underline{x}(0)$ of (6) there exists a control sequence $\underline{u} = -\underline{x}(0)$ taking $\underline{x}$ to 0 at $k = n$, practical limitations such as the accuracy of the initial measurements of $\underline{x}(0)$ and the accuracy of the levels the actual control sequence indicate that as a practical matter, what one can actually achieve is to go from a neighborhood of $\underline{x}(0)$ to a neighborhood of 0. Nothing is lost then by realistically assessing the potentialities of considering controls, $u(k)$, which can assume only discrete levels and using these discrete levels of control to take neigh-

neighborhoods of the initial state to neighborhoods of the origin. Assuming that all possible initial states of (6) are bounded, that is

$$||\underline{x}(0)|| \leq M \quad (9)$$

where $||\cdot||$ is a norm, then it is possible to precisely formulate the problem. To do this, we require a discretization of the state space into $Q+1$ fixed discrete values $\{\underline{x}_j^* ; j=0,1,\dots,Q\}$, a discretization of the level of the control variable into $P+1$ fixed discrete values $\{u_k^* ; k=0,1,\dots,P\}$. For each $\underline{x}_j^*(0)$, there corresponds exactly one control sequence $\underline{u}_j^* = \{u_j^*(0), u_j^*(1), \dots, u_j^*(n-1); u_j^*(i) \in \{u_k^*\}\}$ taking $\underline{x}_j^*(0)$ to the origin, that is $\underline{x}_j^*(k)$ given by (6) is $\underline{x}_j^*(n) = \underline{0}$.

The discrete values $\{\underline{x}_j^*\}$ and $\{u_k^*\}$ are to be chosen such that, given an M and $\epsilon > 0$, a $\delta(\epsilon) > 0$ can be found such that for any initial state $\underline{x}(0)$ with $||\underline{x}(0)|| \leq M$, there exists an $\underline{x}_j^*$ with $||\underline{x}(0) - \underline{x}_j^*|| \leq \delta$, and a control sequence, $\underline{u}_j^*$, corresponding to $\underline{x}_j^*$ such that with $\{u(k) = 0, k = n, n+1, \dots\}$, $\{||\underline{x}(k)|| \leq \epsilon; k = n, n+1, \dots\}$.

Under the assumptions made, regarding (6), such a discretization can always be made.

To see this, define $\tilde{\underline{x}}(0)$ by

$$\tilde{\underline{x}}(0) = \underline{x}(0) - \underline{x}_j^* \quad (10)$$

Selecting the sequence $\underline{u}_j^*$ to be that which drives $\underline{x}_j^*(0)$ to $\underline{0}$ at $k=n$ yields for $k \geq n$

$$\underline{x}(k) = \underline{A}^k \tilde{\underline{x}}(0) \quad (k \geq n) \quad (11)$$

Since $|\lambda_i(A)| < 1$, there exists a finite number m defined by

$$m = \sup_{j=1,2,\dots} ||\underline{A}^j|| \quad (12)$$

such that

$$||\underline{x}(k)|| \leq m ||\tilde{\underline{x}}(0)|| \quad (13)$$

Choosing $\delta(\epsilon)$ to be

$$\delta(\epsilon) < \epsilon/m \quad (14)$$

yields

$$||\underline{x}(k)|| \leq \epsilon \quad (k \geq n; ||\underline{x}(0) - \underline{x}_j^*|| \leq \delta) \quad (15)$$

It should be noted that if $\underline{A}$ is contractive, that is, $||\underline{A}|| \leq 1$, then $m \leq 1$ and δ could be chosen as $\delta = \epsilon$.

Since the set $\{\underline{x}; ||\underline{x}|| \leq M\}$ is bounded it can be covered by a finite number of neighborhoods of the type $\{||\underline{x} - \underline{x}_j^*|| \leq \delta; j = 0,1,2,\dots,Q\}$. This proves that the discretization is possible. It is clear that Q must depend on an M, ϵ and $||\cdot||$, $Q = Q(M, \epsilon, ||\cdot||)$.

NORM SELECTION

The selection of the norm determines the topology of the state space and the control space. The development of the preceding section did not actually depend on the canonical form specified in (6,7,8). It is this canonical form that makes the selection of the norm convenient. Although any norm could be used on the basis of the preceding development, there might not then be a unique control sequence taking any neighborhood to a neighborhood of the origin. In particular, some $\underline{x}(0)$ could exist such that $\underline{x}(0)$ is in a neighborhood of two different discrete values, say $\underline{x}_{j1}^*$ and $\underline{x}_{j2}^*$.

By defining the norm to be given by

$$\|\underline{x}\| = \max \{|x_i|; i=1,2,\dots,n\} \quad (16)$$

and defining the δ -neighborhood of a point $\underline{x}$ by

$$N_\delta(\underline{x}) = \{\underline{y}; -\delta < y_i - x_i \leq \delta; i=1,2,\dots,n\} \quad (17)$$

the neighborhoods are half-open (or half-closed) n -dimensional cubes of edge length 2δ and centered at $\underline{x}$. It is easy to see that with this topology, the set $\|\underline{x}\| \leq M$ can be covered by $(2p+1)^n$ neighborhoods where p is the smallest integer larger than M/δ . The centers of the neighborhoods then occur at vectors, $\underline{x}$, with coordinates x_i given by

$$x_i = \pm 2k_i\delta \quad (k_i = 0,1,\dots,p; i = 1,2,\dots,n) \quad (18)$$

Let X represent this set of $(2p+1)^n$ vectors

$$X = \{\underline{x}; x_i = \pm 2k_i\delta; k_i = 0,1,\dots,p; i = 1,2,\dots,n\} \quad (19)$$

From (7) and (17), the control sequence $\underline{u}^*$ consisting of that sequence taking the neighborhood center vector $\underline{x}_{j1}^*(0) \in X_j$ to zero consists of a sequence of n discrete values $\underline{u}_j^* = \{u_j^*(0), u_j^*(1), \dots, u_j^*(n-1)\}$ each of which take on exactly one of $2p+1$ possible discrete values, that is; $u_j^*(q) \in D$ ($q=0,1,\dots,n-1$) where D is the set of $2p+1$ numbers given by

$$D = \{\pm 2k\delta; k = 0,1,\dots,p\} \quad (20)$$

This same control sequence takes any $\underline{x}(0) \in N_\delta(\underline{x}_{j1}^*(0))$ to the neighborhood $N_\epsilon(\underline{0})$, that is

$$\{\underline{x}(k); k \geq n\} \subset N_\epsilon(\underline{0}) \quad (21)$$

A ROUTING ANALOGY

There is an obvious routing analogy which may be made. Let $\underline{u} = \text{col}\{u(0), u(1), \dots, u(n-1)\}$, and consider any discretized version of the problem. To take an $\underline{x}(0) \in N_\delta(\underline{x}_{j1}^*)$ to $N_\epsilon(\underline{0})$, there are $(2p+1)^n$ distinct possibilities for $\underline{u}$, that is, in going from the vertex, $u(k)$, to the vertex $u(k+1)$, there are $2p+1$ different choices or paths for each $k = 0,1,\dots,n-1$. If a cost matrix is associated with the problem to constrain the sequential route selected to take $\underline{x}_{j1}^*(0)$ to $\underline{0}$, that is, the optimal route is $\underline{u}_j^* = -\underline{x}_{j1}^*(0)$, then the route $\underline{u}_j^* = -\underline{x}_{j1}^*(0)$ conditioned by taking $N_\delta(\underline{x}_{j1}^*)$ to $N_\epsilon(\underline{0})$ and by

allowing the $u_j^*(q)$ to be only $u_j^*(q) \in D$, is a feasible route. However, for some $\underline{x}(0) \in N_\delta(\underline{x}_j^*)$, there may exist a $\hat{u} \neq \underline{u}_j^*$ with $\hat{u}(q) \in D$ but such that $\{\underline{x}(k); k \geq n\} \subset N_{\varepsilon_1}(0)$ where $\varepsilon_1 < \varepsilon$. Thus the optimum routing problem is to find the $\hat{u}^*$ which minimizes ε_1 subject to the constraint $\hat{u}^*(k) \in D$ for $k = 0, 1, \dots, n-1$.

Practically, a feasible route is obtained by determining for any $\underline{x}(0)$ with $||\underline{x}(0)|| \leq M$, the $\underline{x}_j^*$ such that $\underline{x}(0) \in N_\delta(\underline{x}_j^*)$ or in other words, find the n values of k_i such that

$$k_i \delta < x_i(0) \leq (k_i + 2) \delta \quad (k_i = \pm 1, \pm 3, \dots; i=1, \dots, n) \quad (22)$$

Then the corresponding $\underline{x}_j^*$ has the coordinates

$$x_{j,i}^* = (k_i + 1) \delta \quad (i = 1, \dots, n) \quad (23)$$

and the feasible route then is given by

$$u(q) = - (k_{q+1} + 1) \delta \quad (q = 0, \dots, n-1) \quad (24)$$

If $\underline{x}(0) = \underline{x}_j^*$, from (6, 7, 8) and the $u(q)$ of (24), it follows that

$$\underline{x}(q+1) = \underline{H} \underline{x}(q) \quad (q = 0, 1, \dots, n-1) \quad (25)$$

where $\underline{H}$ is given by

$$\underline{H} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (26)$$

It is clear that $\underline{x}_j^* \in X$ implies $\underline{H}^q \underline{x}_j^* \in X$ for $q=0, \dots, n-1$. Hence, if $\underline{x}(0) \neq \underline{x}_j^*$ and $\underline{x}(0) \in N_\delta(\underline{x}_j^*)$, it may turn out that for some q with the control determined by (24) that $\underline{x}(q)$ given by

$$\underline{x}(q) = \underline{H}^q \underline{x}_j^* + \underline{A}^q \tilde{\underline{x}}(0) \quad (27)$$

is

$$\underline{x}(q) \notin N_\delta(\underline{H}^q \underline{x}_j^*) \quad (0 < q \leq n-1)$$

or

$$\underline{x}(q) \in N_\delta(\underline{x}_p^*) \neq N_\delta(\underline{H}^q \underline{x}_j^*)$$

Thus it may be more advantageous to select a new control sequence based on the present state, $\underline{x}(q)$. It should be noted that if $\underline{A}$ is a contraction mapping, then $\underline{x}(0) \in N_\delta(\underline{x}_j^*)$ implies from (27) that $\underline{x}(q) \in N_\delta(\underline{H}^q \underline{x}_j^*)$.

These concepts are intimately related to the techniques of Neighborhood Storage [5, 6, 7] and it is felt that these should be useful in further examining this theory.

REFERENCES

1. Vogt, W. G., Gunderson, R. W., "Canonical Realizations," presented at the Asilomar Conference on Circuits and Systems, November, 1967 and published in the Proceedings.
2. Vogt, W. G., "Multi-Input Systems Controllable with a Single Input," presented at the 1968 SWIEEEO Conference, Houston, Texas, April, 1968; published in the Proceedings, pp. 19 D 1-2.
3. Vogt, W. G., Cullen, C. G., "The Minimal Number of Inputs Required for the Controllability of a Linear Stationary Dynamical System." IEEE Trans. on Automatic Control, Vol. AC-12, No. 3, June, 1967, p. 314.
4. Vogt, W. G., Mickle, M. H., "Canonical Form for Deadbeat Control," ELECTRONICS LETTERS, vol. 4, no. 6, March, 1968, pp. 102-103.
5. Mickle, M. H., Sze, T. W., Rathbone, D. E., "Neighborhood Storage," IEEE Transactions on Systems Science and Cybernetics, Vol. 4, No. 2, pp. 138-144, July, 1968.
6. Mickle, M. H., Duda, F. T., "Abstract Formulation and Increased Efficiency of Neighborhood Storage," To be published.
7. Duda, F. T., Mickle, M. H., "Optimum Vertex Selection and Neighborhood Determination for Neighborhood Storage," To be published.

C.L. HENDRICKS A.J. KOIVO S.J. CITRON
Purdue University, Lafayette, Indiana

ABSTRACT

A production-inventory problem is studied in a form amenable to the discrete-time optimal control theory. First, the mathematical model for the system is assumed to be described by a set of linear difference equations with time-varying coefficients. A quadratic cost function for the system is optimized with respect to the production capacity using dynamic programming. Then, a general mathematical model is presented for the production-inventory problem. The discrete-time maximum principle is applied to optimize the system. Examples for a production-inventory system are presented. The interpretation of the results is given in a form which can serve as a guideline for decisions by management.

INTRODUCTION

This paper studies the production-inventory problem in the framework of system theory. The production-inventory problem can be considered from the viewpoint of discrete-time optimum control theory as follows. Management has a choice of values for certain decision variables, for example, the size of the work force. Once management has made these decisions there are various constraints which determine the production output and the inventory level; for example, production level is usually constrained to increase in proportion to the size of the work force. These constraints form a set of input-output relations which, depending on the particular situation, may be called the plant dynamics. These input-output relations convert the management decisions (input) into finished goods (output). To optimize the physical system, it is first necessary to determine a cost function or performance index for the system.

The purpose of stocking a warehouse is to insure that products will be available when customers demand. The amount of inventory effects such costs as obsolescence, handling, damage, insurance, and interest on the capital which is tied up in the goods stored. Each of these inventory carrying costs can be decreased by keeping fewer items on hand, but this action will increase the expected cost of not being able to fill the customers' orders. This stockout option decreases profit not only because the customer may fill his order elsewhere, but also because he may continue to fill his orders elsewhere if he receives such poor service very often. The amount of inventory also has an effect on production scheduling. For example, if there is a large demand for goods and only a small inventory, then most probably a significant increase in production would occur to meet this demand to avoid a large stockout cost. Such changes in production level usually create costs beyond those of constant production. If there is an increase in production then overtime and hiring costs are added, while a sudden decrease creates idle time and layoff costs. Thus it is desirable to find the production-inventory policy which minimizes the cost function. The policy found in this manner is termed the optimal production-inventory

* Supported in part by Chrysler Foundation and in part by National Science Foundation Grant No. GK-1970.

policy.

A common approach to the production-inventory problem appearing in the literature is to assume that both the constraint equations and the cost function are linear and then to find a solution by linear programming [2,3,4]. The production-inventory problem has also been studied in the framework of quadratic cost functions by means of dynamic programming [5,8]. However, the models for the systems studied are quite restricted. It is hoped that the presentation here will help to achieve a more descriptive model. For this purpose, dynamic programming is used in the determination of an optimal policy for a system described by a multidimensional set of linear constraint equations and a quadratic cost function both having time-varying coefficients. Also a non-linear production-inventory problem is investigated by stating necessary conditions for an optimal solution and by computing a numerical solution.

A. SPECIAL CASE: LINEAR PLANT-QUADRATIC CRITERION

To start the study, a system governed by a set of linear difference equations is considered. The system is assumed to be a sampled-data process with a uniform sampling interval of Δt , and the minimization of the cost function is performed over a finite time. The variables to be considered are: the production level, the deviation of inventory from some desired inventory level⁺; the sales level; and the production capacity, which constitutes the control variable. The sales function is assumed to be a known piecewise linear function of time.

Symbolically the above quantities are defined as follows⁺⁺:

$$\begin{aligned} u(i) &= \text{capacity for production decided upon at time } i\Delta t, i=0,1,\dots \\ x_1(i) &= \text{production level at time } i\Delta t. \\ x_2(i) &= \text{difference between actual and desired inventory at time } i\Delta t. \\ I(i) &= \text{inventory level at time } i\Delta t. \\ I^* &= \text{desired inventory level; assumed to be a known constant.} \\ d(i) &= \text{sales level at time } i\Delta t. \end{aligned} \quad (1)$$

The interval between $i\Delta t \leq t < (i+1)\Delta t$ is denoted here as the i^{th} time period. According to the definition of $x_2(i)$ the following relation holds:

$$x_2(i) = I(i) - I^* \quad (2)$$

It is assumed that the production level rises exponentially to the capacity u . The rate of increase is determined by the constant c according to the relation:

⁺ "Desired inventory level" is that level which optimally balances the costs due to carrying inventory and the costs due to not filling a demand for goods. That is, it is a function of the desired level of service and is sometimes called safety stock or "cushion" inventory [7].

⁺⁺ In the sequel the subscripts indicate the components of a vector.

$$x_1(i+1) = x_1(i) + c[u(i) - x_1(i)] \quad (3a)$$

or combining terms

$$x_1(i+1) = (1-c)x_1(i) + cu(i) \quad (3b)$$

It is reasonable to assume that the sales during period i are subtracted from the inventory at time $(i+1)\Delta t$, whereas the production during period i is added to the inventory. Therefore the following relation can be written (the conservation law of inventory):

$$I(i+1) = I(i) + [x_1(i) - d(i)]\Delta t \quad (4a)$$

In the sequel, the interval Δt is taken to be one unit. Subtracting I^* from both sides in equation (4a) leads to

$$I(i+1) - I^* = I(i) - I^* + x_1(i) - d(i) \quad (4b)$$

Substituting equation (2) into equation (4b) yields

$$x_2(i+1) = x_2(i) + x_1(i) - d(i) \quad (5)$$

A compact form for equation (3) and (5) can be obtained by defining

$$x_1(i) = \begin{bmatrix} x_1(i) \\ x_2(i) \end{bmatrix} \quad u(i) = [u(i)] \quad A(i) = \begin{bmatrix} 1-c & 0 \\ 1 & 1 \end{bmatrix} \quad (6)$$

$$B(i) = \begin{bmatrix} c \\ 0 \end{bmatrix} \quad \text{and} \quad D(i) = \begin{bmatrix} 0 \\ -d(i) \end{bmatrix}$$

Then equations (3) and (5) become

$$x(i+1) = A(i)x(i) + B(i)u(i) + D(i) \quad (7)$$

Next, the cost is to be formulated in terms of the production level, the inventory deviation, and the production capacity. For the model used here, the cost associated with negative production (self-consumption) is assumed to be the same as that associated with positive production: the cost of carrying excess inventory is the same as that associated with being below the desired inventory level. The cost of changes in production level and the cost due to production capacity are not considered here. With these restrictions, the total cost is assumed to be

$$J[u] \equiv \bar{p} x_1^2(N) + \bar{q} x_2^2(N) + \sum_{i=1}^{N-1} \{p x_1^2(i) + q x_2^2(i)\} \quad (8)$$

where N is the total number of time periods considered, and $\bar{p}$, $\bar{q}$, p , q are specified positive constants (weighting factors).

It is noted that the initial value of the production level and the initial value of the inventory deviation do not contribute to the cost function (since the initial values are assumed specified). Also, because the process is over at time $N\Delta t$, the charge for $x(N)$ is usually different from the charge for $x(i)$, $i = 1, \dots, N-1$. Equation (8) can be rewritten in a more compact form

$$J[u] = \sum_{i=0}^{N-1} \{x'(i+1)Q(i+1)x(i+1) + u'(i)\Gamma(i)u(i)\} \quad (9)$$

where

$$Q(i) = \begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix}; \quad Q(N) = \begin{bmatrix} \bar{p} & 0 \\ 0 & \bar{q} \end{bmatrix}; \quad \text{and } \Gamma(i) = [0]$$

and the superscript $(\cdot)'$ indicates the matrix transposition; $i=1, \dots, N-1$.

The charge for $\{u(i)\}$ is included in $J[u]$ for generality. In expression (9), $Q(i)$ is positive definite and $\Gamma(i)$ is at least positive semi-definite for all i . Under these assumptions, the problem is to minimize $J[u]$ in equation (9) subject to the constraint equations (7). The optimization problem stated by equation (7) and (9) is obviously valid in a higher dimensional case as well.

A1. Solution by Means of Dynamic Programming. The method of dynamic programming [6] can be successfully used to solve the linear plant-quadratic criteria problem specified by equations (7) and (9). The minimum of the pay-off function is defined in the usual manner:

$$V(N-j)[x(j)] = \text{minimum value of the cost of the } (N-j)\text{-stage process from state } x(j). \quad (10)$$

Then the application of the principle of optimality yields

$$V(N-j)[x(j)] = \min_{u(j)} \left\{ x'(j+1)Q(j+1)x(j+1) + u'(j)\Gamma(j)u(j) + V(N-(j+1))[x(j+1)] \right\} \quad (11)$$

Equations (9) and (11) give $V_0[x(N)] = 0$. The pay-off function can be written in the form

$$V(N-i)[x(i)] = x'(i)P(N-i)x(i) + 2x'(i)R(N-i) + C(N-i) \quad (12)$$

where $P(N-i)$ is an $n \times n$ matrix; $R(N-i)$, an n -vector; and $C(N-i)$, a scalar; recursive relations can be determined for $P(N-i)$, $R(N-i)$, and $C(N-i)$.

Carrying out the minimization in (11) and using (12) produces the optimal control law:

$$u^*(i) = -K(i)x(i) - \bar{K}(i) \quad (13)$$

where $K(i) = \eta(i)^{-1}B'(i)SA(i)$; $\bar{K}(i) = \eta(i)^{-1}B'(i)[SD(i) + R(N-(i+1))]$;

$$S = Q(i+1) + P(N-(i+1)); \quad \eta(i) = B'(i)SB(i) + \Gamma(i).$$

Substituting (13) and (12) into (11), the needed recursion relations for $P(N-i)$ and $R(N-i)$ are obtained.

$$P(N-i) = A'(i) \left[I - SB(i) \eta(i)^{-1}B'(i) \right] SA(i) \quad (14)$$

$$R(N-i) = A'(i) [I - SB(i) \eta(i)^{-1} B'(i)] (SD(i) + R(N-(i+1))) \quad (15)$$

where the boundary conditions are specified by $C(0) = 0$; $R(0) = 0$; $P(0) = 0$. Equations (14) and (15) are used to generate the sequences of $\{P(N-i)\}$ and $\{R(N-i)\}$. The closed loop optimal control law is then determined by equation (13).

A2. Example 1. For illustration, the dynamic programming method is applied to the problem expressed by equations (7) and (9). The sales function to be used corresponds to one that might occur in a company introducing a new product, where the product is disposed of after using. Also, it is assumed that the company is planning to replace this product on the market at some known future time. For this example, the product is taken to have a sales function which increases from zero up to the market saturation value, and then remains constant until the item is taken off the market. The only costs are assumed to be those due to the production level and the deviation of the inventory from the desired level. The cost of production is assumed to be five times greater than the cost of storing. Thus it is desired to find the solution to the problem of minimizing

$$J[u] = \sum_{i=1}^N \{5x_1^2(i) + x_2^2(i)\} \quad (16)$$

$$\begin{aligned} \text{subject to: } x_1(i+1) &= (1-c)x_1(i) + cu(i) \\ x_2(i+1) &= x_1(i) + x_2(i) - d(i) \end{aligned} \quad (17)$$

where c is the production constant; it is assumed to be 0.5.

The quantities x_1 , x_2 , u and d are defined by equation (1) with $d(i) = i$ for $0 \leq i \leq 10$ and $= 10$ for $i > 10$ and $x'(0) = [0, 0]$.

For this specific example, $Q(i) = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$; $\Gamma(i) = [0]$ for every i

and $\bar{p} = p = 5$; $\bar{q} = q = 1$ in equation (8).

The optimal results are found by applying the equations (13)-(17). Figure 1 displays the solution. The solution for the inventory deviation indicates that excess inventory should be stored during the first periods, and production alone should be used to meet the required demand. In the latter periods the demand should be met in ever-increasing amounts from inventory and constantly decreasing amounts from production.

A3. Planning Horizon. In the situation described in Section A2, it might happen that the time originally scheduled for replacement of the product on the market is later extended. Thus an optimal production-inventory policy would need to be found for the period between the time at which this decision is reached and the new termination time decided upon. In order to determine the effect of extending the marketing period, the optimal policy for the process when the total marketing life is correctly predicted is compared to the optimal policy when the marketing life is extended once and when it is twice extended. It is assumed that the time at which it is decided to extend the marketing period is the time originally

decided upon as the replacement is denoted by N . For the case of three marketing subintervals the intermediate times at which the decisions are made are denoted by N_1 and N_2 . A problem of this type can be solved with the approach presented.

The results are presented in Figures 2-4. For this example, when the process is extended once, the cost over the entire process $[0, N]$ is greater than in the case that the planning for the actual market life, including extensions, is done at the beginning of the process (cf. Figure 2). If the process is extended twice the situation is even more costly as can be seen in Figure 3. This increase in cost can be explained as follows. When management considers the shorter replacement time, the optimal policy is to produce so as to have a large deficiency in inventory at the time of replacement. This results in a bad initial situation for the extended marketing period. At the new starting point, the optimal policy requires management to eliminate the inventory deficiency rapidly. Such an action results in a production level higher than the optimal, and the higher production level increases the cost. To illustrate the argument, the results for the production level and the inventory deviation are shown in Figure 4 for the case of three marketing subintervals. The costs found due to extensions in marketing lifetime indicate that for the production-inventory problem described by equations (7) and (8) it is worth a considerable amount of time and money to determine the correct lifetime of the product.

B. GENERALIZATION WITH LIMITED PRODUCTION CAPACITY

The policy for the production capacity in the problem may be subject to physical limitations. For example, management may not be able to hire as many workers or obtain as many machines as would be necessary to achieve the desired production capacity. In such a case, the results obtained in the absence of constraints are no longer optimal. Also, if the relations between production and inventory are non-linear or if the cost is not a quadratic function of production and inventory, the dynamic programming solution presented is no longer directly applicable. Instead of extending the dynamic programming method to such a general problem, an alternative approach is introduced. That is, necessary conditions for an optimal solution can be written from the discrete-time maximum principle. The resulting equations pose a two-point boundary-value problem which is usually difficult to solve explicitly. However, a numerical solution can be obtained by means of a digital computer.

B1. Necessary Conditions for an Optimal Solution. The general production-inventory problem has the following form:

$$\underset{\{u\}}{\text{minimize}} J[u] = f_N(x(N)) + \sum_{i=0}^{N-1} f_i(x(i), u(i)) \quad (18)$$

$$\text{subject to: } x(i+1) = F(x(i), u(i)), x(0) \text{ specified} \quad (19)$$

where $i = 0, 1, \dots, N-1$; $u(i) \in U$, U is convex set; $x \in E^n$,

$F \in C^2$ is vector valued, $f_i \in C^2$ for all $i=0, 1, \dots, N$.

Necessary conditions for a local minimum are written by means of the discrete-time maximum principle [1]: If the sequence $\{u^*(i)\}$ $i = 0, \dots, N-1$

and $\{x^*(i)\}$ $i = 0, 1, \dots, N$ are optimal then there exists a sequence of non-zero vectors $\{\lambda_i\}$, $i = 1, \dots, N$ such that

$$\lambda'(N) = \frac{\partial f_N(x(N))}{\partial x(N)} \quad (20a)$$

$$\lambda'(i) = \lambda'(i+1) \frac{\partial F(x^*(i), u^*(i))}{\partial x(i)} + \frac{\partial f_i(x^*(i), u^*(i))}{\partial x(i)} \quad (20b)$$

$$\frac{\partial}{\partial u(i)} \left[f_i(x(i), u(i)) + \lambda'(i+1) F(x(i), u(i)) \right] \begin{matrix} \delta u(i) \geq 0 \\ u^*(i) \\ x^*(i) \end{matrix} \quad (20c)$$

for every $\delta u(i)$ such that $u^*(i) + \delta u(i) \in U$

$$u^*(i) \in U \quad (20d)$$

$$x^*(i+1) = F(x^*(i), u^*(i)) \quad (19)$$

$$\text{where } \frac{\partial f_i(x(i), u(i))}{\partial x(i)} = \left[\frac{\partial f_i(x(i), u(i))}{\partial x_1(i)}, \dots, \frac{\partial f_i(x(i), u(i))}{\partial x_n(i)} \right]$$

$$\frac{\partial F(x(i), u(i))}{\partial x(i)} = \begin{bmatrix} \frac{\partial F_1(x(i), u(i))}{\partial x_1(i)}, \dots, \frac{\partial F_1(x(i), u(i))}{\partial x_n(i)} \\ \vdots \\ \frac{\partial F_n(x(i), u(i))}{\partial x_1(i)}, \dots, \frac{\partial F_n(x(i), u(i))}{\partial x_n(i)} \end{bmatrix}$$

Since in general these equations are difficult to solve explicitly, a gradient technique, say, can be used to determine a candidate for the optimal solution iteratively.

B2. Example 2. Consider the following production-inventory problem

$$\underset{u}{\text{minimize}} J[u] = \bar{p} x_1^2(N) + \bar{q} x_2^2(N) + \sum_{i=1}^{N-1} \{p x_1^2(i) + q x_2^2(i)\} \quad (21)$$

$$\text{subject to: } x_1(i+1) = (1-c)x_1(i) + cu(i) \quad (22a)$$

$$x_2(i+1) = x_2(i) + x_1(i) - d(i) \quad (22b)$$

and $|u(i)| \leq u_{\max}$ where $u_{\max}$ is a given constant, and x_1, x_2, u , and d are defined by (1). For this specific example $d(i) = i$ for $0 \leq i \leq 10$ and $= 10$ for $i > 10$; and $x'(0) = [0, 0]$. Necessary conditions on the co-state variables and the optimal control are:

$$\lambda_1(N) = 2\bar{p} x_1(N); \quad \lambda_2(N) = 2\bar{q} x_2(N) \quad (23a)$$

$$\lambda_1(i+1) \delta u(i) \geq 0 \text{ implies } u^*(i) = u_{\max} \operatorname{sgn} [\lambda_1(i+1)] \quad (23b)$$

$$\lambda_1(i) = (1 - c)\lambda_1(i+1) + \lambda_2(i+1) + 2p x_1(i) \quad (23c)$$

$$\lambda_2(i) = \lambda_2(i+1) + 2q x_2(i), \quad i = N-1, \dots, 1 \quad (23d)$$

Equations (22) and (23) form a discrete-time two point boundary value problem. In order to obtain a solution, a gradient method [9] is employed.

This method consists of assuming a certain set of $\{u(i)\}$, $i = 0, \dots, N-1$ and then changing this set by $\{\delta u(i)\}$ in order to decrease the value of $J[u]$. It is accomplished by forcing δJ negative in each step while limiting the value of δu . Then the process is repeated. The bound on $|u(i)|$, namely $u_{\max}$, is handled by simply setting $u(i) + \delta u(i)$ equal to $u_{\max}$ if $u(i) + \delta u(i)$ exceeds $u_{\max}$ and equal to $-u_{\max}$ if $u(i) + \delta u(i)$ is less than $-u_{\max}$. The iteration procedure is then repeated until the value of $J[u]$ cannot be improved anymore. Since the method converges to a possible local solution, the iteration process should be started with various initial approximations.

The results of the computational solution for this example are shown in Figure 5. The bound $u_{\max}$ was given the values 2, 4, 6, 8, and 10. For values of 10 and greater, the bound on the production capacity is greater than the maximum value for any time period of the unbounded optimal control sequence; thus the resulting solution to (21-23) is the same as the results given in Section A2. When the bound on $u_{\max}$ is small - 2, 4, and 6 - the solution indicates that production is not capable of meeting the demand in later periods. As a result, the inventory deviation during the later periods contributes significantly to the cost function. Consequently, the management policy in case of the smaller bounds should be to call for maximum capacity for every time period, but the very last one. For a bound of 8 on the production capacity, however, the optimal production policy is to request less than the maximum in the first periods.

DISCUSSION

Although the examples treat the problem with a linear-plant quadratic criterion, the approach is applicable to a general problem as well; for example, in case the sales function is chosen to be a more complicated function. Moreover, nonlinear terms in the model of the plant can readily be handled by the method presented. It is also noted that it has been assumed in the problems studied that the state variables can take on any real values. In many practical situations, however, the production level and inventory deviations assume only integer values. It is believed that the resulting quantization error is not too severe, particularly when the numbers under consideration are large.

CONCLUSIONS

It is shown here how the discrete-time optimal control theory can be applied to minimizing the cost of a production-inventory system. Dynamic programming yields useful results for a system described by a multidimensional set of linear constraint equations and a quadratic cost function both having time-varying coefficients. Also a general mathematical model is presented for the production-inventory problem. The resulting system is optimized by applying the discrete-time maximum principle. Numerical results are presented and interpreted from the viewpoint of the management.

REFERENCES

1. Halkin, H., Jordan, B.W., Polak, E. and Rosen, J.B., "Theory of Optimum Discrete Time Series", MRC Technical Summary Report #601, Sept. 1965.
2. Hanssman, F. and Hess, S.W., "A Linear Programming Approach to Production and Employment Scheduling", Management Tech., Monograph No. 1, p. 46-51, Jan. 1960.
3. Veinott, A.F., "Production Planning with Convex Costs: A Parametric Study", Management Science, Vol. 10, No. 3, p. 441, Apr. 1964.
4. Lippmann, S.A., Rolfe, A.O., Wagner, A.M. and Yuan, J.S.C., "Algorithms for Optimal Production Scheduling and Employment Smoothing", Operations Research, Vol. 15, No. 6, p. 1011, Nov.-Dec., 1967.
5. Holt, C.C., Modigliani, F., Muth, J.F. and Simon, H.A., Planning Production, Inventories, and Work Force, Prentice Hall Inc., Englewood Cliffs, N.J., 1960.
6. Tou, J.T., Modern Control Theory, San Francisco, McGraw-Hill, 1964.
7. Elmaghraby, S.E.A., "On the Feedback Approach to Industrial Systems Design", Management Science, Models, and Techniques, Vol. 1, New York, Pergamon Press, 1960.
8. Simon, H.A., "Dynamic Programming Under Uncertainty with Quadratic Criterion Function", Econometrica, Vol. 24, No. 1, p. 74, Jan. 1956.
9. Ho, Y.C. and Brentani, P.B., "On Computing Optimal Control with Inequality Constraints", SIAM Journal on Control, Vol. 1, No. 3, Series A, p. 319, 1963.

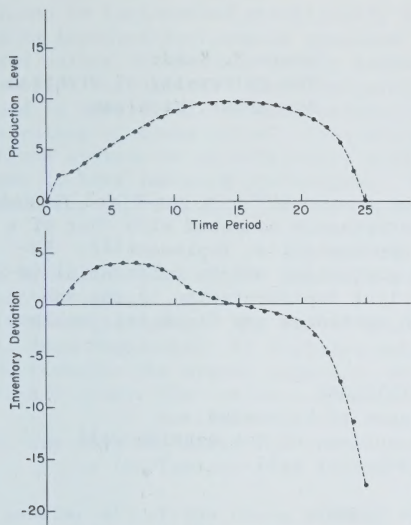


FIGURE 1. OPTIMAL POLICIES IN EXAMPLE 1.

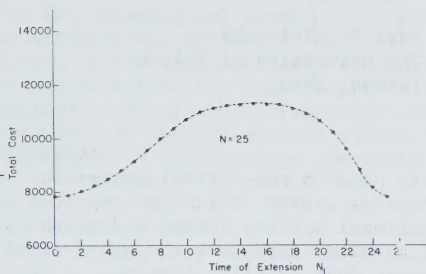


FIGURE 2. EFFECT OF EXTENDING MARKETING LIFE ONCE.

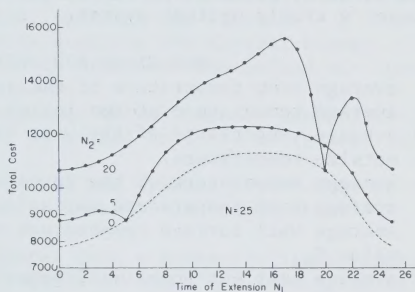


FIGURE 3. EFFECT OF EXTENDING MARKETING LIFE TWICE.

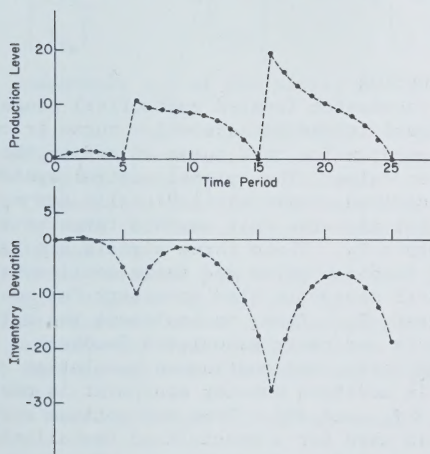


FIGURE 4. SUB-OPTIMAL PRODUCTION AND INVENTORY POLICIES.

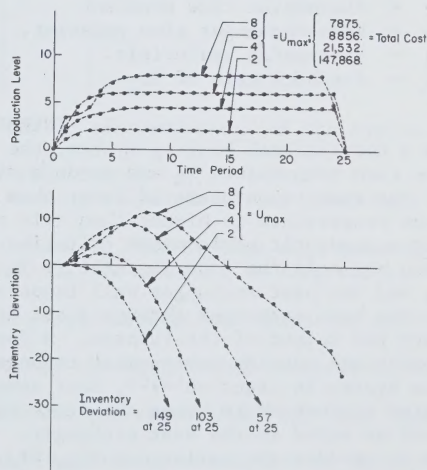


FIGURE 5. OPTIMAL POLICIES: LIMITED PRODUCTION CAPACITY EXAMPLE 2.

A SUB-OPTIMAL HEATING SYSTEM

Adel H. Eltimsahy
The University of Toledo
Toledo, Ohio

Louis F. Kazda
The University of Michigan
Ann Arbor, Michigan

ABSTRACT

In this paper a sub-optimal controller is presented for a gas-fired forced-air heating system which has improved performance compared with that of a conventional heating system and which is economical to implement(1). Results obtained show that the sub-optimal controller offers substantial improvement over existing systems. A practical implementation of the sub-optimal control system is presented which overcomes the financial problems that beset a truly optimal system.

NOTATIONS AND DEFINITIONS

- T_R = average room temperature of the space to be heated.
 T_w = average temperature of the inside surface of the outside wall.
 T_e = average temperature of the heat exchanger wall.
 T_o = outside temperature.
 T_f = average temperature of the furnace flame.
 δT_R = average room temperature variation from some equilibrium value T_{R_o} .
 δT_w = average wall surface temperature variation from some equilibrium value T_{w_o} .
 δT_e = average heat exchanger wall temperature variation from equilibrium value T_{e_o} .
 δT_f = average flame temperature variation from the equilibrium value T_{f_o} .
Cycling Amplitude = peak to peak room temperature when an on-off controller is used.
Cycling Period = time of one cycle of room temperature when an on-off controller is used.
 τ_T = thermostat time constant
 τ_F = heat exchanger time constant.
 q = thermostat hysteresis.
 k_w = feedback gain of T_w .

I. INTRODUCTION

In a conventional heating system, the thermostat (system controller) senses the room temperature T_R and sends a signal to the furnace which turns it on if the room temperature is lower than a set value, and turns it off if the room temperature is higher than this set value. The optimal control system for a quadratic performance criterion derived previously(1,6) calls for sensing not only the room temperature T_R but also the wall surface temperature T_w and the heat exchanger wall temperature T_e . These three signals are fed to the heat exchanger through a set of feedback gains and these continuously vary the output of the furnace. A fourth sensor is also necessary for monitoring the outside environment temperature T_o . Thus, to implement the optimum system in practice(1,6), four sensors and their associated feedback gains are needed to sense the state variables, and continuous modulation must be added to the heat exchanger. In addition summing equipment is needed to produce the variations δT_R , δT_w , δT_e , and δT_f . Thus the optimum system is in general expensive unless it is used for a specialized installation with rigid performance standards which are independent of cost. For domestic heating it would be particularly costly due to the additional expenses attributable to a continuously modulated furnace, four temperature sensors, and the associated circuitry making the optimum system impractical. Therefore, it is logical to attempt to devise a sub-optimum heating system, which

(1) can be implemented practically, (2) is acceptable economically, and (3) yields improved performance compared to the conventional system. The proposed method of approach mainly trades mathematical optimality in return for practical usefulness. The structural form of the feedback gains are determined in an optimal fashion, meanwhile computer runs are used to prespecify the values of these gains. In a comparison study of the conventional and optimum systems on an electronic analog computer the sub-optimal system was shown to have improved performance. The paper is concluded by describing a method for implementing the sub-optimal controller.

II. THE SYSTEM PLANT

The fixed portions of the domestic heating system must include the following basic elements:

1. Domestic Space: The space the temperature of which is to be controlled
2. Room Boundaries: It includes walls, ceiling, and floor.
3. Furnace: The system component which supplies the thermal energy.
4. Air Ducts: The system component which transfers the heated air throughout the habitable enclosure.
5. Gas Control Valve: The system component which releases combustible gas to the furnace heat exchanger on signal from the controller unit.

In general all of the above elements affect the dynamic performance of the system; effects of some of the components however, are negligible⁽¹⁾.

The above listed components are integrated into a schematic block diagram indicating their relationship to the overall system. Such a diagram is shown in Figure 1.

An approximate mathematical model for the fixed components of the system has been established⁽¹⁾ and is expressed by the following single vector linear differential equation: $\dot{x} = Ax + u + m$ where:

$$x = \begin{bmatrix} T_R \\ T_w \\ T_e \end{bmatrix}, u = b_3 \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, m = d_2 \begin{bmatrix} 0 \\ T_o \\ 0 \end{bmatrix}, A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & 0 & a_{33} \end{bmatrix}$$

The components a_{ij} of the matrix A, b_3 , and d_2 are parameters of the system.

III. PROPOSED SUB-OPTIMUM SYSTEM

The principal properties of the optimum heating system developed previously^(1,6) may be summarized as follows:

- (a) Three state signals, the room temperature T_R , the surface wall temperature T_w , and the heat exchanger temperature T_e are combined to form the control signal.
- (b) Three feedback gains are required to provide the proper weighting of the state signals T_R , T_w , and T_e .
- (c) An external sensor is necessary to sense the environment temperature T_o , and generate the equilibrium values.
- (d) Summing equipment is employed to generate the variations δT_R , δT_w , δT_e , and δT_f .
- (e) A heat exchanger with the capacity for continuous modulation is required.

For a conventional heating system the main properties can be summarized as follows⁽¹⁾:

- (a) A single signal, the room temperature T_R , is used for control purposes.
- (b) A heat exchanger with only an on-off characteristics is used.

The characteristics of any practical sub-optimum heating system which are proposed should draw heavily from the properties of the optimum heating system. To develop a sub-optimal system incorporating the properties of the optimal and conventional systems, the following reasonings are used:

(i) In present day forced-air gas heating systems a heat exchanger with continuous modulation equipment is considerably more expensive than one without such equipment. The improvement in the system due to adding such equipment versus the cost of the equipment is not acceptable at the present time to the consumer. Therefore, it is essential that the sub-optimum system utilize an on-off heat exchanger.

(ii) In both conventional and optimum heating systems room temperature T_R is sensed, and controlled. Therefore, it follows that room temperature T_R should be controlled in any sub-optimum system which is proposed.

(iii) In the optimum heating system, the wall temperature T_w , is sensed and utilized for control purposes. This means that the T_w sensor attempts to predict environment phenomena, by sensing the wall surface temperature, and uses this prediction to send a signal to the furnace to react before significant changes occur in the room temperature. Intuitively this principle can be applied in a sub-optimum system. Therefore, it is suggested here that the sub-optimum system include a sensor to monitor T_w and feedback a signal to improve variations in the room temperature. T_{w0} the equilibrium temperature at the wall surface is set at a value corresponding to a room temperature of 70°F and the average outside temperature under consideration(1).

(iv) The heat exchanger temperature T_e sensed in the optimum system should not be sensed in the sub-optimum for economical reasons.

For the reasons stated above the proposed sub-optimum system has the following main characteristics.

- (a) Heat is applied from a heat exchanger in an on-off fashion.
- (b) The room temperature T_R is sensed and fed back for comparison with a reference signal.
- (c) The surface wall temperature T_w is sensed and fed with an appropriate gain factor.

This proposed sub-optimum heating system is shown schematically in Figure 2.

IV. ANALYSIS OF THE SUB-OPTIMUM HEATING SYSTEM

The sub-optimum system described in Section III is simulated on the electronic analog computer for the purpose of investigating and analyzing its performance. The fixed portions of the heating system described in Section II were used in order to compare the two controllers; conventional, and sub-optimum.

In this investigation, a habitable room space of 12'x12'x10' is used(1), and the heat output of the furnace is adjusted so that the temperature of the air circulating in the heating system is 120°F when it is leaving the furnace during the on-period. During the off-period the temperature of the air is considered to be 70°F. The outside temperature T_o is held fixed at 20°F, and then allowed to drop suddenly to zero. Computer runs were made for the system with room air temperature T_R feedback alone, and then with both room air temperature T_R and surface wall temperature T_w feedback. Let k_w be the feedback gain for T_w . In each run, the room air temperature response cycling amplitude and period were recorded. Computer runs were made for different T_w feedback gains. For each of these runs, the thermostat time constant, heat exchanger time constant, and the thermostat hysteresis were varied and plots of the cycling amplitude and period versus the feedback gain of T_w were obtained for various sets of parameters. The plots of these investigations are shown in Figures 3 through 8.

In Figure 3 three curves are displayed. These curves indicate the dependence of the cycling amplitude on the feedback gain of T_w for three values of thermostat hysteresis, q , and for a thermostat time constant, τ_T , of 0.4 minutes and furnace time constant, τ_F , of 1.6 minutes. An examination of the curves shows that for the range of feedback gain considered, the cycling amplitude decreases as a function of feedback gain and then begins to either increase slightly or become constant. The case of a zero feedback gain for T_w represents the conventional controller. For a hysteresis value of 0.18°F the cycling amplitude decreased from 0.63°F to 0.33°F when the controller was changed from a conventional form to a sub-optimal form with a unity feedback factor. This indicates an improvement of about 50% in the room temperature deviation. Figure 4 displays the corresponding curves for the cycling period. It reveals that by starting with a higher value of thermostat hysteresis, the cycling period decreases with feedback gain, and the rate of decrease of this period drops with the thermostat hysteresis. For the case of a zero hysteresis the cycling amplitude tends to increase slightly after it reaches a minimum.

Figure 5 displays three curves showing the cycling amplitude versus the feedback gain for different values of heat exchanger time constant. The thermostat time constant was kept at 0.4 minutes and its hysteresis at 0.12°F . Examination of the curves indicates that for heat exchangers of time constant 5 minutes, 2.5 minutes, and 1.6 minutes, the cycling amplitude is a decreasing function of the feedback gain of T_w . For values of feedback gain $k_w > 0.9$ the cycling amplitude begins to be constant. Therefore, when a heat exchanger of a time constant of 5 minutes is used with the heating system and a conventional controller, it shows a 0.657°F cycling amplitude and 0.55°F cycling amplitude when used with the suggested sub-optimal controller with 0.725 gain, i.e. it shows an improvement of approximately 19% when the sub-optimal controller is used. Figure 6 shows the corresponding cycling period set of curves. At lower values of heat exchanger time constant, the cycling period decreases gradually, and as the heat exchanger time constant increases, the room temperature cycling period decreases and subsequently begins to increase at a very small rate.

Figure 7 displays a graph indicating the cycling amplitude versus the feedback gain, when the thermostat time constant is varied as a parameter. The heat exchanger time constant was held constant at 5 minutes, and the thermostat hysteresis at 0.12°F . These curves further demonstrate that the room temperature cycling amplitude is a decreasing function of gain, until $k_w > 0.81$ where it starts to level off. Figure 8 shows the corresponding cycling period behavior.

Therefore, it can be stated generally that the room temperature cycling amplitude decreases as the feedback gain of the surface temperature T_w increases and it levels off at $k_w > 0.8$. The cycling period is consistently within practical values i.e. within cycling period values which do not lead to working the mechanical parts of the heating system at a very high frequency. It is to be noted at this point that the practical implementation of a gain different from unity requires more circuitry. Thus, it is recommended for economical reasons to use $k_w = 1$ for the proposed sub-optimum controller.

Reviewing the analysis of these graphs indicates a definite improvement in the sub-optimum system over the conventional heating system. This improvement is apparent from the differences in the various room temperature cycling amplitudes. Therefore, it may be concluded that the addition to the conventional heating system of a feedback signal which is proportional to the tem-

perature of the wall surface, improves the performance of that system.

If either the sub-optimal or conventional system is to be compared with the optimal system, the basis of comparison must be the quadratic performance criterion. It is true by definition that the optimal system is the best with respect to this criterion; however, interesting points can be made by analyzing the systems in general.

If the conventional heating system is analyzed and compared to the optimum heating system it is found that⁽¹⁾:

(i) the peak to peak variations of the room temperature are much greater for the conventional heating system, when compared to the maximum deviation of the optimum system.

(ii) For an outside temperature disturbance the response and adjustment of the optimum heating system is superior to the corresponding response of the conventional heating system^(1,6). In the optimum system, the temperature begins to fall gradually (due to an outside temperature drop) until it deviates to -0.15°F . Then within about 5 minutes it tends to remain to within 0.1°F or less from the original value. The conventional heating system, on the other hand, begins to oscillate. The rates of increase and decrease in the room temperature depend on the thermostat time constant, and thermostat hysteresis. They also depend on the nature of the disturbance.

(iii) The rate of change in the room temperature is greater when using the conventional controller thus causing the conventional heating system to be less comfortable.

From the analog computer results on the sub-optimal system, it is evident that this system lies in-between the conventional controller and the optimum controller as far as general characteristics are concerned.

(i) The room temperature variations are lower in the conventional system, but higher than in the optimal system; however, the maximum room temperature variations are closer to the optimum system.

(ii) The sub-optimum system oscillates between on and off positions when a disturbance is applied, a characteristic of the conventional system. This is anticipated since an on-off furnace system is employed in the sub-optimum system.

(iii) The sub-optimum controller tends in general to have a smaller cycling period; however, these values are by no means near to operating the heating system at a high mechanical frequency which leads to excessive wear on the gas-valve.

V. THE PRACTICAL IMPLEMENTATION OF THE SUB-OPTIMAL CONTROLLER

It can be shown that⁽¹⁾ a preferred on-off heating system is one possessing a small thermostat time constant, a small thermostat hysteresis value, and a small heat exchanger time constant. The time constant of the heat exchanger depends primarily on its size, which in turn is determined by the size of the system to be heated. The time constant of the thermostat on the other hand, depends primarily on the properties of the bimetal and the speed of its response. One method of improving the time constant of the sensing device is to employ a thermistor. Thermistors vary in response according to quality which is directly proportional to cost. For example, thermistors of the type Keystone RL21FIT have about 20 seconds time constant, and are priced at a very reasonable rate. Such a time constant is much smaller than the time constant of any bimetallic thermostat. The amount of hysteresis of a bimetallic thermostat depends primarily on the properties of the relay or the switching device used. Physically, a switching device or relay will produce some hysteresis. However, this hysteresis may be minimized by increasing the sensitivity of the system. The sensitivity of such a thermo-

stat system can be increased by amplifying the signal that is proportional to the room temperature deviation.

Therefore, taking these ideas into consideration along with the fact that the addition of a feedback signal that is proportional to T_w improves the performance of the heating system, the sub-optimal controller can be implemented practically at a reasonable cost. The idea of this sub-optimal controller is sketched in Figure 9. The temperatures T_R and T_w are sensed by thermistors in the sub-optimal controller, rather than by using a conventional bimetal element. These signals are compared with reference signals T_{R_0} and T_{w_0} , by using an electrical bridge circuit. The difference signal is then amplified by a one stage transistor amplifier. The output of the amplifier actuates a relay. This relay then operates a gas valve of the furnace in an on-off manner. The electrical bridge, the transistor amplifier, and the relay circuits all receive power from the supply through a transformer which must be included. Figure 10 shows the detailed solid state sub-optimal controller.

The feedback gain, k_w , for the surface wall temperature, T_w , is selected as unity in this design. This selection allows for the use of fewer components than if a system possessing non-unity gain was used, and at the same time, it renders considerable improvement to the system.

The room temperature reference T_{R_0} is set by the resistor R_3 . T_{w_0} , the wall surface temperature reference, is set by R_4 . Thermistor R_5 senses the actual room temperature T_R , and thermistor R_9 senses the actual surface wall temperature T_w . The output of the bridge circuit which has a signal proportional to $(\delta T_R + \delta T_w)$, feeds into a transistor amplifier Q_1 . The output of this amplifier operates the relay K_1 , which in sequence turns the gas valve on or off.

Preliminary experiments on the sub-optimal controller revealed a 75% reduction in the time constant from the average existing bimetallic thermostat. The hysteresis of this sub-optimal controller, had a value of 0.18°F^* . The best bimetallic thermostat available showed approximately 1°F hysteresis in the same test.

SUMMARY

In this paper a sub-optimal controller for the gas-fired, forced-air heating system has been presented which has improved performance compared with that of a conventional heating system, and which also has an economical practical implementation. This system was simulated on the analog computer and its performance compared to the optimal system and the conventional system. After showing that the sub-optimal controller offers a substantial improvement over existing systems, a practical implementation was presented and shown to overcome the financial problems which beset a truly optimal system.

*These values were measured using the standard chamber of the Maxitrol Company, Southfield, Michigan. The authors express their thanks to Maxitrol Company for making available their facilities.

REFERENCES

1. "The Optimization of Domestic Heating Systems," Adel H. Eltimsahy, Ph.D. dissertation, April, 1967, University of Michigan.
2. "The Optimal Control of Gas-Fired Warm Air Heating Systems," by L. F. Kazda, G. Casserly, A. Eltimsahy and R. Spooner - AGA Final Report Contract DO-14-GU, University of Michigan.
3. Optimization Theory and the Design of Feedback Control Systems, C. W. Merriam III, McGraw-Hill, 1964.
4. "The Warmed-Air Domestic Space Heating System considered as a closed loop control system," by M. Burrows and L. F. Kazda - ORA Final Report, project 03506, University of Michigan.
5. Modern Control Theory, J. Tou, McGraw-Hill, 1964.
6. "An Optimal Heating System," Adel H. Eltimsahy and L. F. Kazda. Submitted for publication at the 1969 IFAC, Congress IV.

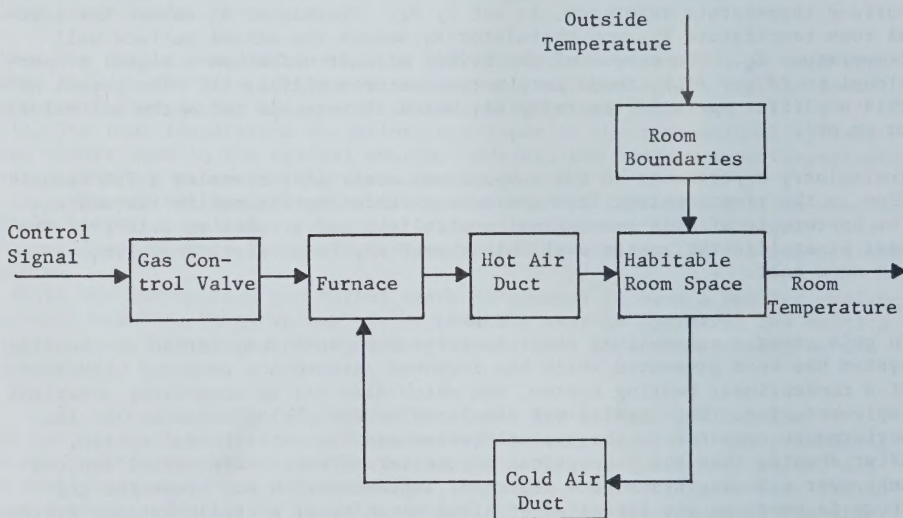


Figure 1. Components of the Heating System

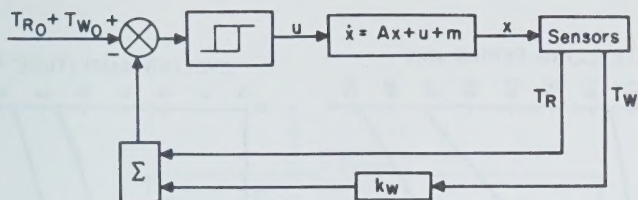


FIGURE 2. BLOCK DIAGRAM SCHEMATIC OF THE SUB-OPTIMUM HEATING SYSTEM.

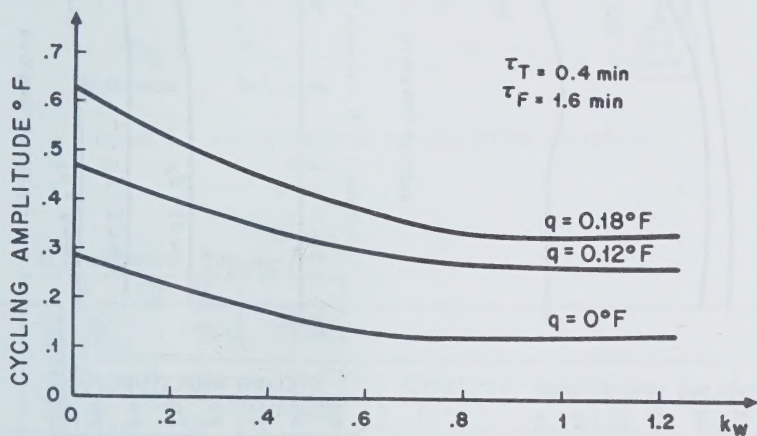


FIGURE 3. CYCLING AMPLITUDE- k_w PLOTS WITH q AS A PARAMETER.

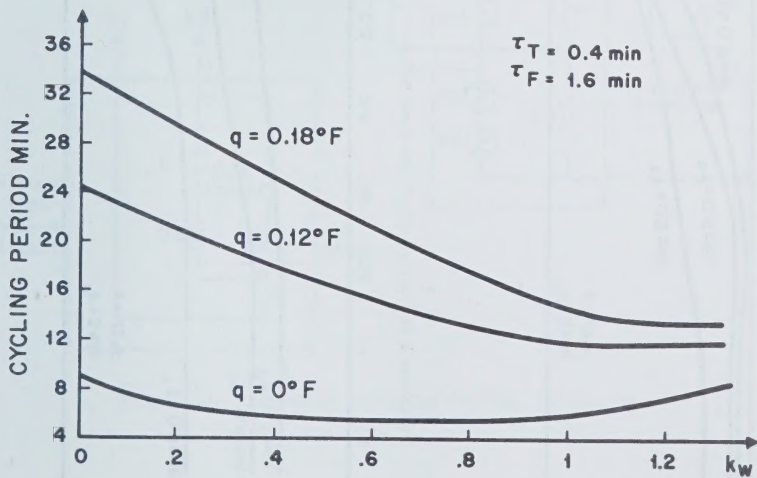


FIGURE 4. CYCLING PERIOD- k_w PLOTS WITH q AS A PARAMETER.

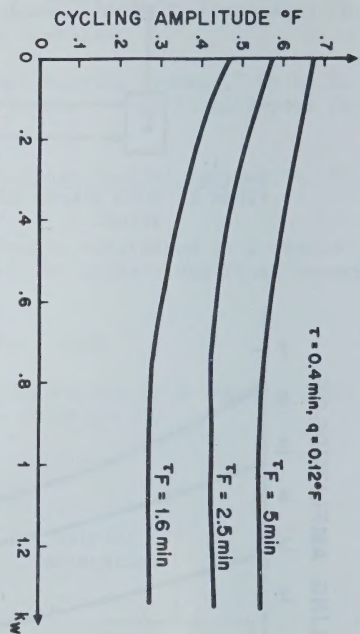


FIGURE 5. CYCLING AMPLITUDE- k_w PLOTS WITH T_F AS A PARAMETER.

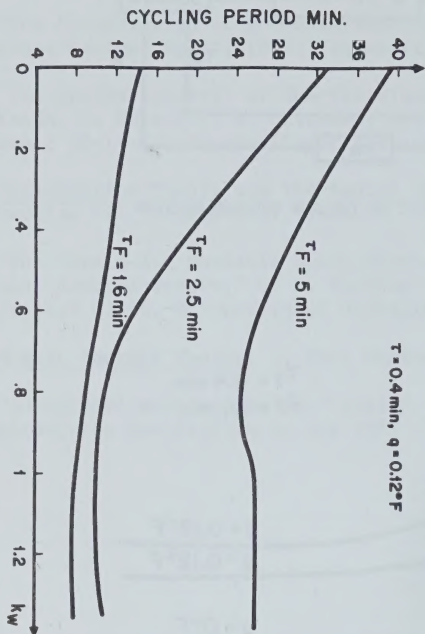


FIGURE 6. CYCLING PERIOD- k_w PLOTS WITH T_F AS A PARAMETER.

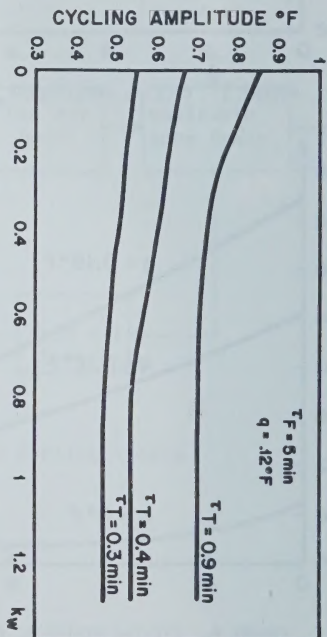


FIGURE 7. CYCLING AMPLITUDE- k_w PLOTS WITH T_T AS A PARAMETER.

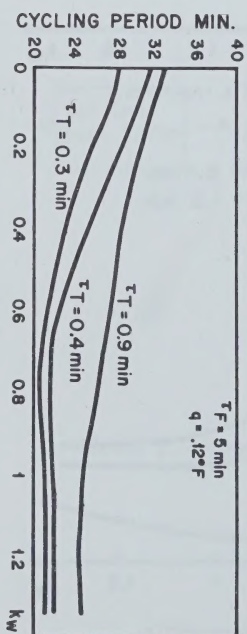


FIGURE 8. CYCLING PERIOD- k_w PLOTS WITH T_T AS A PARAMETER.

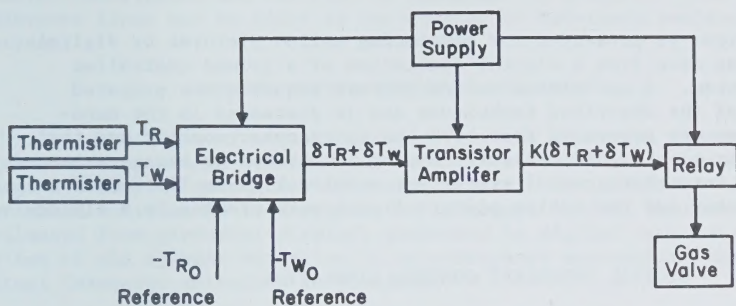


FIGURE 9. BLOCK DIAGRAM OF THE SUB-OPTIMAL CONTROLLER.

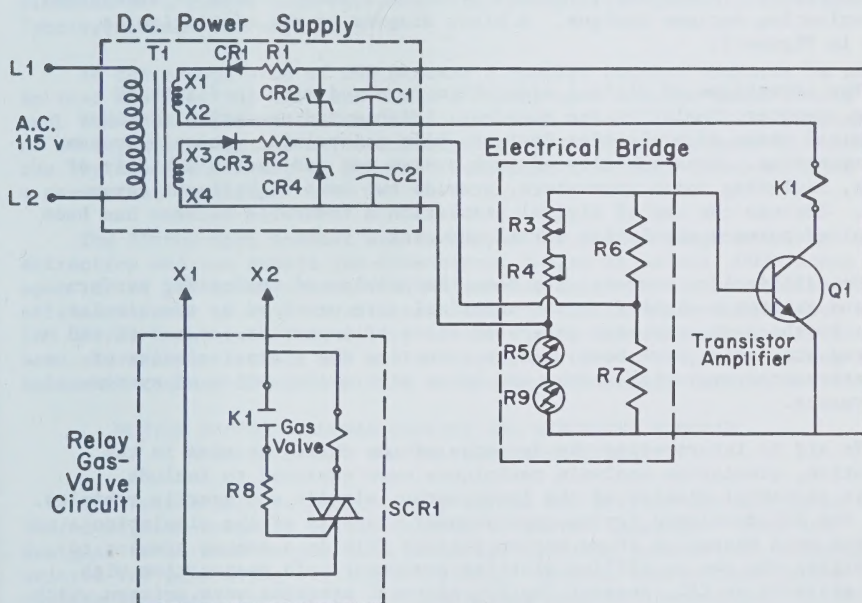


FIGURE 10. THE SOLID STATE SUB-OPTIMAL CONTROLLER.

ANALYSIS OF INTERCEPT CONTROL TECHNIQUES WITH COMPUTER GENERATED MOVIES

G. S. WILLHOFF, C. W. SWONGER, H. P. SEMMELHACK, R. C. FRIEND
Cornell Aeronautical Laboratory, Inc., Buffalo, New York

ABSTRACT

A technique is presented for generating motion pictures by digital computer using data from a digital simulation of a ground-controlled intercept system. A one-minute motion picture sequence was prepared through use of the described techniques and is discussed in the manuscript. Views are presented from both the interceptor cockpit and the head of an air-to-air guided missile during an attempt to engage a manned target. The intercept control system was simulated on an IBM 360/65 digital computer and the motion picture frames were created by a flying-spot scanner.

DIGITAL INTERCEPT CONTROL SIMULATION

The digital computer simulation of a complete intercept control system was developed at Cornell Aeronautical Laboratory, Inc. for studies of ground-controlled intercepts. The simulation was designed with several primary objectives, including: high speed operation, accurate representation, and flexibility of operation to simulate a wide range of system configurations. A modular structure provides instant change of subsystems for evaluating various designs. A block diagram of the simulation is shown in Figure 1.

The advantages of digital simulation compared with in-flight or analog computer simulation are numerous. High speed operation provides economical means of evaluating designs while maintaining accurate system representation. Repeatability of each system run and controllability of inputs, including noise generators, provide two more significant advantages. Through the use of digital simulation a favorable balance has been maintained between simulation detail and cost.

One difficulty, however, has been the problem of evaluating performance and gaining insight from the numerical data produced by the simulation. In the past, computer generated plots of important parameters and selected statistics have been used to summarize the characteristics of the interceptor vectoring system and as an aid to evaluation of system performance.

To aid in interpreting the dynamics of the three vehicles in the simulation, simulation analysis techniques were expanded to include dynamic pictorial display of the interceptor, missile and hostile vehicles. Using the CAL-developed flying-spot scanner, records of the simulation dynamics were traced on 16-mm motion picture film in a manner similar to controlling the pen on offline plotting equipment. In conjunction with other projects at CAL, several "motion picture" programs were written which use the basic plotting package to draw objects in perspective at locations on the motion picture frames specified by the simulation. The objects themselves consist of line segments specified as data to the motion picture package. Movement of the objects is suggested when the simulation updates dynamic variables at each frame interval.

Numerous variations of motion picture techniques are possible with the present motion picture package. The user has the ability to create slow or fast motion sequence. One may choose to keep the camera stationary or it may be removed at will in the user's program. Various special effects are possible such as zooming in on a particular detail or alphanumeric description and labeling of items in the picture. Grid and/or reference lines may be added to the display to reference vehicular motions.

COMPUTER MOTION PICTURE GENERATION¹

The generation of motion pictures from the intercept control simulation outputs provides a unique and valuable research tool for evaluating large and complex quantities of research data. For the first time all variables of interest to designers of intercept control systems may be evaluated from pictorial displays generated by digital computer. Observation of all dynamic variables in an atmosphere approaching that of the actual intercept situation is possible.

A block diagram of the motion picture generation technique is shown in Figure 2. Two steps are required in the actual motion picture making process. First, the Fortran-coded motion picture programs are used to generate a plotting tape from data supplied by the intercept control simulation. The motion picture generating programs implement all the elements of a camera-lens system and the perspective geometry involved in "photographing" the objects.

In the second step of the process a special purpose computer is used to read the plotting tape and drive a flying-spot scanner which traces each frame on a strip of 16-mm motion picture camera film. For purposes of debugging, a digital X-Y coordinate pen plotter may be used in place of the flying-spot scanner to generate hard copies of each frame enlarged by a convenient factor.

The flying-spot scanner was developed for use in image property extraction and can sample two-dimensional images at points .003 inches apart in 64 levels of gray at 40,000 samples/sec. It can be used as an offline scanner as well as with scan control and input or output from an IBM 360/65 computer. The utility of the flying-spot scanner has recently been extended to include its use as a graphic output device by adding an automatic-control 35-mm recording camera and developing control software.

MOTION PICTURE PROGRAM PACKAGE AND EXECUTIVE PROGRAM FOR INTERCEPT CONTROL MOTION PICTURES

The motion picture generating programs are written for use on an IBM 360/65 digital computer in the Fortran IV source language. They are designed to be used in the same manner as plotting subroutines commonly written for generating offline X-Y plots. Several subroutines in the package implement the motion picture making process, including: a camera

1. Based upon internally sponsored research at Cornell Aeronautical Laboratory, Inc.

and lens system, routines that draw circles and lines in perspective, a routine that sets up new motion picture frames, and routines that initiate or delete objects during the filming process. All objects are defined as an array of points prior to execution of the calling program. During a call to the package subroutines, an object is "drawn" or "filmed" with position and attitude specified by the executive program. Figure 3 shows a block diagram of the program.

The executive program for generating motion pictures of intercept control processes allows the motion picture camera to be mounted on either the interceptor or the interceptor's air-to-air missile. The camera position, when fixed to the interceptor, generates a view that would be seen by the pilot when following air controller orders, thus giving a feeling for the severity and accuracy of interceptor vectoring data. When mounted on the nose of an air-to-air missile fired by the interceptor, the resulting film sequence allows performance evaluation with regard to approach angle, missile lags and possible countermeasures which could have altered the effectiveness of the missile against the target. These points immediately become apparent upon viewing the film sequence, whereas, before the method was implemented, much effort was spent in evaluating numerical quantities produced by the simulation.

Although the two camera locations mentioned above have been the only ones used to date, provision has been made, by being sufficiently general in program logic, so that additional scenes and views of an intercept may be produced with slight modification to the program.

SAMPLE MOTION PICTURE SEQUENCE

In Figure 4, a sequence from the one-minute intercept-control film is shown which was produced by the motion picture generation package. Each frame is .05 seconds apart in the terminal phase of an intercept with the camera mounted on the air-to-air missile. The target, shown in the approximate center of the frames is traveling in a direction from left to right, while the missile is attempting a collision course with the target. The exact center of each frame is the true heading or line of sight of the missile. Note that the target appears to travel across the line of sight of the missile, indicating that the missile guidance system lags are producing a lead-pursuit course.

The sequence in Figure 4 was produced on a CALCOMP plotter in an actual dimension of 10 inches by 7.5 inches per image. These are selected images produced from a typical intercept control simulation run.

Figure 5 shows a motion picture sequence which was produced from the same data tape as the graphic output in Figure 4. The sequence of images was generated by the flying-spot scanner in the manner described previously: a computer generated motion picture.

Two views are shown in Figure 5: a simulated pilot's view and a wide angle series of images from an air-to-air missile. Details in the sequence produce a feeling of movement and depth as well as serving as reference points. These objects include stationary clouds above the horizon line and range marks on the earth's surface. This sequence is an example of the usefulness of the simulation in pointing out problems of complicated interceptor-missile guidance designs. The missile performance indicated here was difficult to predict from numerical data but becomes quite obvious in the missile sequence.

This paper describes an initial application for digital computer motion picture generation as a research tool. In addition to generating intercept control simulation motion pictures, the package is currently being used to generate computer motion pictures of single vehicle accidents through the data supplied by CAL's digital computer automobile crash simulator. Future plans include online generation of motion pictures through direct linkage of the IBM 360/65 and the CAL Flying-Spot Scanner, the use of interactive graphics and the development of a motion picture production language. Potential new applications encompass the areas of tracking system investigations, aircraft motion analysis, pattern recognition and data clustering studies, and signal processing research.

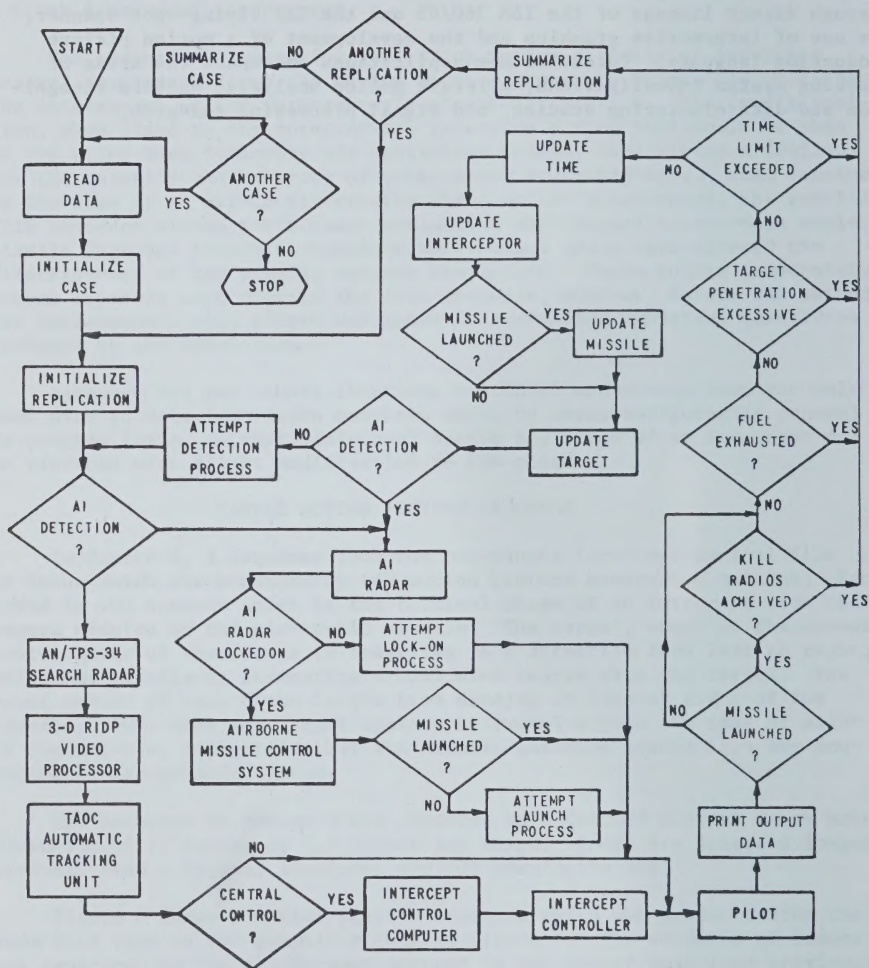
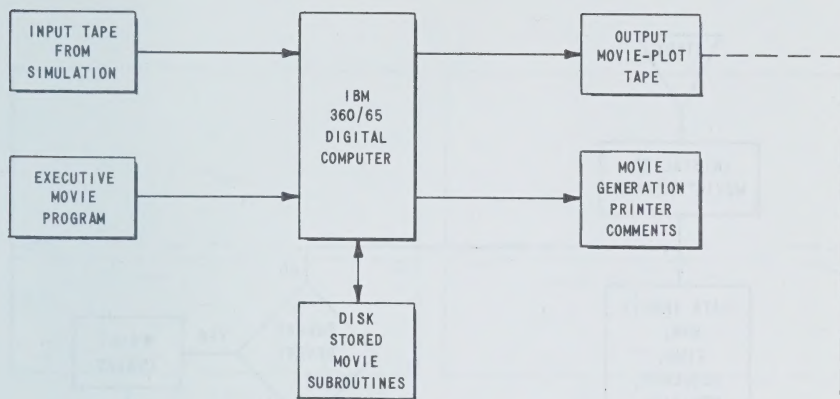
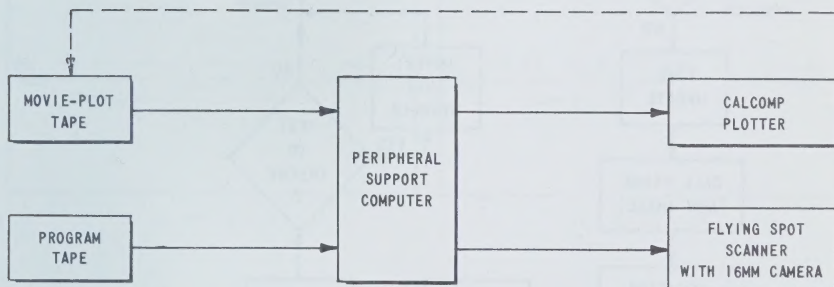


Figure 1 INTERCEPT CONTROL SIMULATION



(a) COMPUTING EQUIPMENT



(b) MOTION PICTURE GENERATION HARDWARE

Figure 2 COMPUTER MOTION PICTURE GENERATION EQUIPMENT

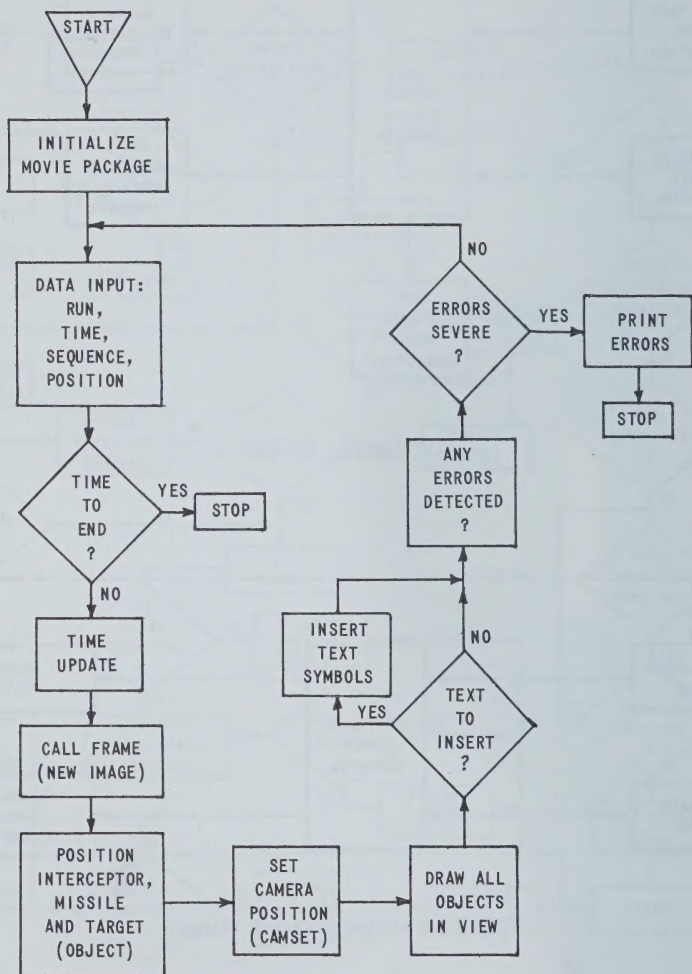
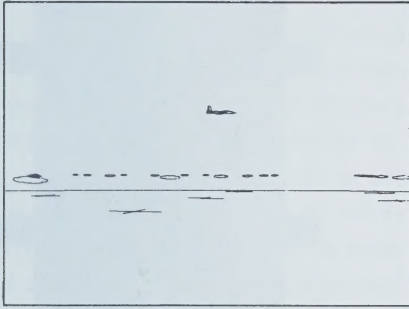
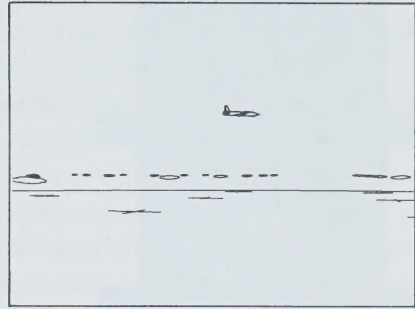


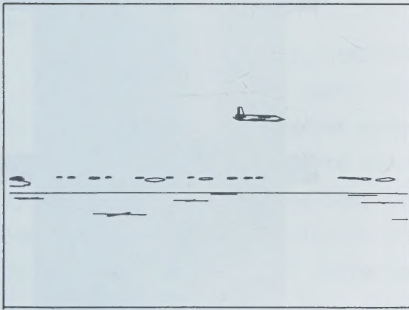
Figure 3 PROGRAM FOR PRODUCING INTERCEPT CONTROL MOVIES



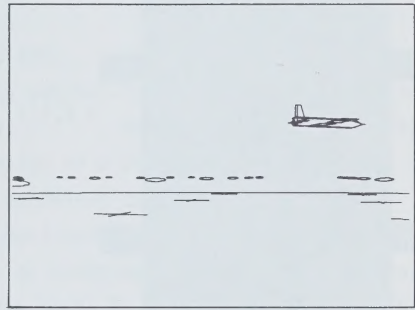
(a) $T = T_0$



(b) $T = T_0 + .05$



(c) $T = T_0 + .1$



(d) $T = T_0 + .15$

Figure 4 TYPICAL GRAPHIC SEQUENCE PRODUCED BY THE
PERSPECTIVE MOTION PICTURE PROGRAM

BLOW-UP OF MOVIE FILM

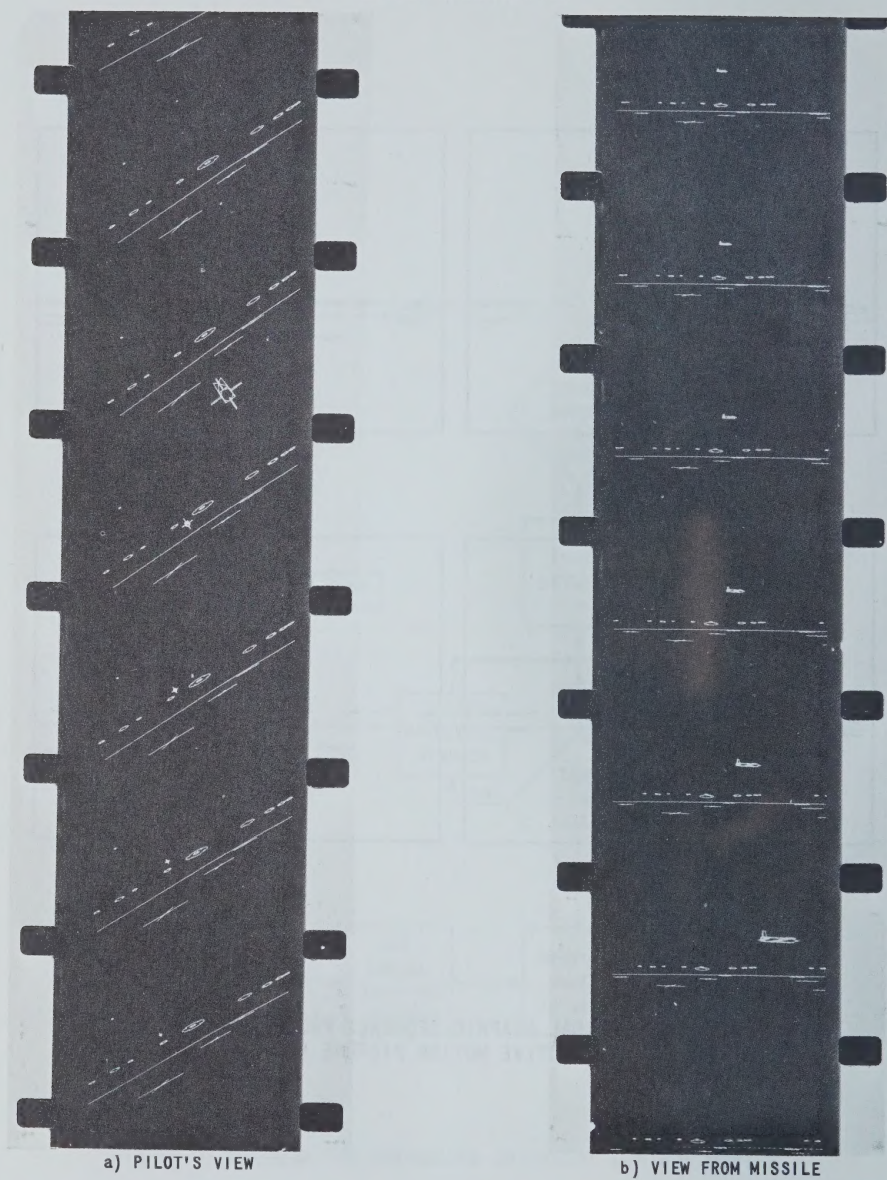


Figure 5 TYPICAL MOTION-PICTURE SEQUENCES

A CLASS OF DISCRETE SIGNAL DESIGN PROBLEMS IN BURST NOISE⁺

Howard I. Silver
Fairleigh Dickinson University
Department of Electrical Engg.
Teaneck, New Jersey

Ludwik Kurz
New York University
Department of Electrical Engg.
University Heights
Bronx, New York

ABSTRACT

In this paper, the design of useful signalling alphabets is approached as a discrete signal design problem. A minimum Gabor bandwidth binary alphabet is formulated as a discrete optimization problem to illustrate the design philosophy. The solution is first obtained through the use of difference equations and then as an eigenvalue problem. The difference equation technique is extended to the design of signals in a burst noise environment, the problem being formulated as a discrete differential game. The eigenvalue approach is also applied to this problem, where the burst is simulated by the removal of samples. Higher order alphabets are constructed for specific sample sizes and their performance evaluated. A "two-player" differential game approach is also presented as an alternate design procedure.

I. INTRODUCTION

The design of optimal pulse shapes subject to certain constraints has been the subject of a number of papers.^{1,2,3,4} Nuttall and Amoroso⁵ have extended some of these ideas to the design of a useful M-ary alphabets subject to energy, orthogonality, and bandwidth constraints. In all of these works, however, the signals are designed on a continuous time basis with the aid of calculus of variations or Pontryagin's maximum principle.

The design of signals on a discrete basis is a useful technique for several reasons. First, the detection scheme employed by a communication system often results in making a decision on a discrete sampled version of the output. The amplitudes at the sampled points, frequently taken at equal intervals, are then of concern rather than a continuous description of the signal. Also, the discrete version of an optimization problem often lends itself to certain recursion techniques leading to computational simplicity. Design flexibility is thus obtained with respect to the continuous analog of the problem. Finally, one can often consider the continuous problem as a special case of the discrete one where the increment between adjacent samples is made arbitrarily small.

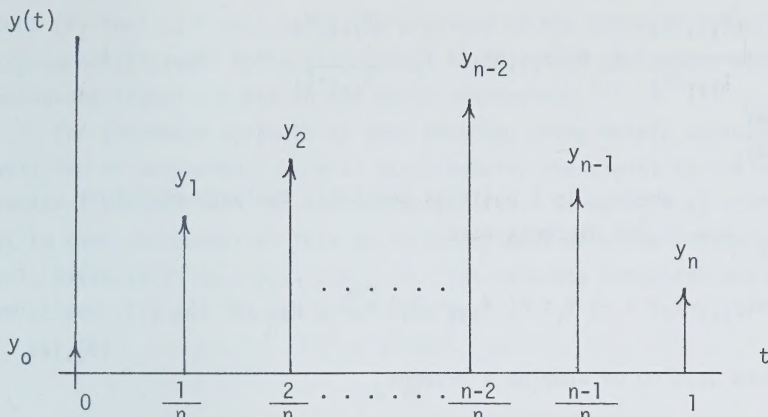
⁺The research in this paper was performed at New York University and supported by the National Science Foundation under Grant GK-1075.

In this paper, the minimum bandwidth pulse subject to energy and end-point constraints, originally considered in its continuous version by Gabor¹, is formulated as a discrete optimization problem. The solution is first obtained through the use of difference equations, this approach being the discrete analog of the continuous form of calculus of variations. A second approach is presented in the form of a matrix optimization problem, the solution being the principal eigenvector of a characteristic Euler-Lagrange matrix equation. The eigenvalues and corresponding eigenvectors are explicitly determined for this case, the higher order ones being useful in the construction of a discrete M-ary alphabet. The matrix approach is then applied to the problem of designing a useful alphabet in a burst noise environment. The burst is simulated by the removal of samples corresponding to a particular burst location, an approach previously suggested by Simon and Kurz.⁶ The eigenvalues and eigenvectors are not determined explicitly for this case, but a closed form relationship is derived from which they can be determined for a specific sample size, burst location, and burst width. Computer results are obtained for two "narrow" burst and two "wide" burst cases. Higher order signalling alphabets are constructed by averaging corresponding eigenvectors for all specific burst locations over an assumed uniform distribution of burst location, and choosing subsets of signals that preserve orthogonality properties.

The difference equation technique is extended to the design of low dimension signalling alphabets in a burst noise environment. The problem is formulated as a discrete analog of a two-player differential game.⁷ The second player's role is to choose an initial condition of the difference equations to simulate burst width. Performance results of various M-ary alphabets in the burst noise environment are included.

II. DESIGN OF DISCRETE MINIMUM BANDWIDTH SIGNAL

The discrete version of the minimum bandwidth pulse design problem is now formulated mathematically. The signal is specified by a row vector $Y = (y_0, y_1, y_2, \dots, y_n)$, where y_i represents the amplitude of the i -th sample. The samples occur at times $t_0, t_1, t_2, \dots, t_n$. It is assumed that the samples are uniformly spaced; also, y_0 and y_n are chosen to be the sample values at normalized times $t = 0$ and $t = 1$, respectively. Under the assumptions of uniform spacing and zero amplitude at the end-points (to avoid discontinuities), we have $t_0 = 0, t_n = 1, t_{i+1} - t_i = \frac{1}{n}$, ($0 \leq i \leq n-1$). The form of the signal is shown on the following page.



FORM OF DISCRETE SIGNAL

The problem is to design Y to minimize a discrete version of the Gabor bandwidth definition¹ subject to a discrete energy constraint and endpoint conditions. The problem is then the following: design Y to minimize

$$B^2 = \sum_{i=0}^{n-1} \left[\frac{y_{i+1} - y_i}{t_{i+1} - t_i} \right]^2 = n \sum_{i=0}^{n-1} (y_{i+1} - y_i)^2 \quad (1)$$

$$\text{subject to the normalized energy limitation } n = \sum_{i=0}^n y_i^2 \quad (2)$$

and endpoint conditions $y_0 = y_n = 0$.

The problem will initially be solved using difference equations. The differential game notation of Isaacs⁷ will be used for convenience, although we are presently considering a "one-player" game or optimization problem. The Kinematic equations to be used are the following:

$$\dot{t} = 1, \dot{y} = \emptyset, \dot{u} = y^2$$

The payoff function is given by $G = \emptyset^2$. The resulting main equation and path equations are shown below:

$$\begin{aligned} V_y &= 2 \sqrt{V_t + V_u y^2} \\ \dot{t} &= 1, \dot{y} = -\frac{1}{2} V_y, \dot{u} = y^2 \\ \dot{V}_t &= 0, \dot{V}_y = -2y V_u, \dot{V}_u = 0 \end{aligned}$$

In the continuous version of this problem $\dot{y}$ and $\dot{V}_y$ indicate forward derivatives of y and V_y with respect to time. In the discrete case they are interpreted as forward differences, defined as

$$\dot{y} = \frac{y_{i+1} - y_i}{t_{i+1} - t_i} = n (y_{i+1} - y_i); \quad \dot{V}_y = \frac{V_{i+1} - V_i}{t_{i+1} - t_i} = n (V_{i+1} - V_i)$$

where $V_i = \frac{\partial V}{\partial y_i}$

Letting $V_u = -\lambda$, where λ is a positive constant, the path equations for y and V_y are in the discrete case

$$n(y_{i+1} - y_i) = -\frac{1}{2} V_i; \quad n(V_{i+1} - V_i) = 2\lambda y_i \quad (3), (4)$$

Due to uniform spacing of samples $\dot{y}$ becomes

$$\dot{y} = n(y_{i+2} - y_{i+1}) = -\frac{1}{2} V_{i+1} \quad (5)$$

Combining equations (3), (4), and (5) the following difference equation in y is obtained:

$$y_{i+2} - 2y_{i+1} + (1 + \lambda_1) y_i = 0 \quad \text{where} \quad \lambda_1 = \left(\frac{1}{n}\right)^2 \lambda, \quad \lambda > 0, \quad y_0 = 0. \quad (6)$$

It can be shown, using Z-transforms for example, that the solution to the difference equation for y_i is

$$y_i = y_1 \sqrt{\frac{1 + \lambda_1}{\lambda_1}} \sin(i\theta) \quad \text{where} \quad \theta = \tan^{-1} \sqrt{\lambda_1}$$

Since $y_n = 0$, the smallest non-trivial solution for θ is $\theta = \frac{\pi}{n}$

Then,

$$\sqrt{\frac{1 + \lambda_1}{\lambda_1}} = \frac{1}{\sin\left(\frac{\pi}{n}\right)}$$

and the solution is

$$y_i = y_1 \frac{\sin\left(\frac{i\pi}{n}\right)}{\sin\left(\frac{\pi}{n}\right)} \quad (1 \leq i \leq n) \quad (7)$$

The constant y_1 is determined from the energy constraint. It can be seen

from (7) that as n gets large the envelope of the discrete solution approaches the half cycle sine pulse, which is the continuous minimum bandwidth signal subject to the given constraints.¹

The alternate approach to this problem, using matrix techniques, will now be presented. We will again specify the signal by the row vector Y and define Y^T as the transpose of Y . Since $y_0 = y_n = 0$, however, it is more convenient at this point to define Y as a row vector of order $n-1$, given by $Y = (y_1, y_2, \dots, y_{n-1})$. The discrete bandwidth and energy relations, (1) and (2) can be written as

$$B^2 = n^2 YAY^T \tag{8}$$

$$n = YY^T \tag{9}$$

where A is given by

$$A = \left[\begin{array}{cccccc} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ 0 & -1 & 2 & \dots & 0 \\ \vdots & & & \ddots & \\ \vdots & & & & \ddots \\ 0 & 0 & 0 & \dots & 2 \end{array} \right] \left. \vphantom{\begin{array}{c} 2 \\ -1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{array}} \right\} n - 1$$

Note that A is square and tridiagonal.

The problem is now to find Y such that $B^2 = n^2 YAY^T$ is minimized subject to the constraint $n = YY^T$. Since ordinary variational techniques can be applied to matrix equations we can use the usual Lagrange multiplier technique to handle the energy constraint. The object then is to minimize a new function G with respect to Y where G is defined as

$$G = B^2 - \lambda_1 (YY^T) = n^2 YAY^T - \lambda_1 YY^T \tag{10}$$

A necessary condition for the minimization of G is that $\frac{\partial G}{\partial Y} = 0$

Differentiating, we obtain

$$\frac{\partial G}{\partial Y} = 2n^2 AY^T - 2 \lambda_1 Y^T = 0 \text{ or } n^2 AY^T = \lambda_1 Y^T \tag{11}$$

We see that if $\frac{\lambda_1}{n^2}$ is chosen to be an eigenvalue of A , then Y^T which satisfies (11) is $\frac{\lambda_1}{n^2}$ the corresponding eigenvector. It is therefore desirable to find a procedure which determines the $n-1$ eigenvalues of the matrix and the corresponding eigenvectors. It is well known that in variational problems the smallest of the eigenvalues leads to a minimum

solution for the functional involved. In this case, then, it is the eigenvector corresponding to the minimum eigenvalue which will minimize the bandwidth functional expressed by (8). It is also known that the $n-1$ eigenvectors form an orthogonal set, resulting in a useful M -ary signalling alphabet.

It is to be first noted that the matrix A can be expressed as the difference between two matrices as follows: $A = 2I - A'$ (12)

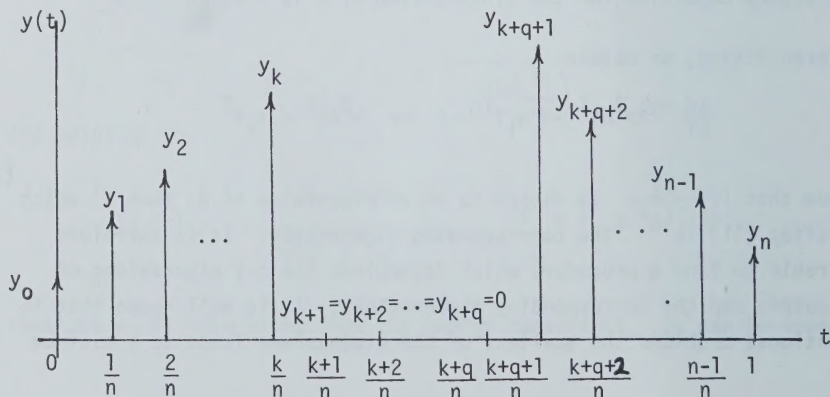
where I is the unit matrix and A' is of order $n-1$ and consists of zeroes along the main diagonal and unity along the diagonals just above and below the main one. It can be easily shown that the corresponding eigenvectors of A and A' are identical. It is more convenient to work with A' since it contains more zero terms than does A , and recursion relations are more easily found. If λ is an eigenvalue of A' then it must satisfy the equation $\det(A' + \lambda I) = 0$. It is noted that as a result of the transformation (12) it is now the largest eigenvalue rather than the smallest which leads to a minimum bandwidth solution. Omitting the details, it can be shown that this eigenvalue is given by $\lambda = 2 \cos(\frac{\pi}{n})$, and the components of the corresponding eigenvector Y are given by

$$y_i = y_1 \frac{\sin\left(\frac{i\pi}{n}\right)}{\sin\left(\frac{\pi}{n}\right)} \quad (1 \leq i \leq n-1) \quad (13)$$

which verifies the result obtained using the discrete form of differential games (7).

III. Discrete Signal Design in Burst Noise Using the Matrix Approach

The discrete signal design problem in a burst noise environment will now be considered. Consider the discrete signal below. This is simply the discrete signal shown previously with a block of samples removed where the burst noise is assumed to occur.



FORM OF DISCRETE SIGNAL IN BURST NOISE ENVIRONMENT

Note that k and q , which define burst location and width, are integers. The problem is then one of designing $Y = (y_0, y_1, \dots, y_n)$ with samples $y_{k+1}, y_{k+2}, \dots, y_{k+q}$ removed. The bandwidth definition (1) must be modified to take into account the removed samples. The following definition will be used:

$$B^2 = \sum_{i=0}^{k-1} \left[\frac{y_{i+1} - y_i}{t_{i+1} - t_i} \right]^2 + \left[\frac{y_{k+q+1} - y_k}{t_{k+q+1} - t_k} \right]^2 + \sum_{i=k+q+1}^{n-1} \left[\frac{y_{i+1} - y_i}{t_{i+1} - t_i} \right]^2 \tag{14}$$

Since the sampling is assumed uniform outside the burst interval

$$t_0 = 0, t_n = 1, t_{i+1} - t_i = \frac{1}{n}, \begin{matrix} 0 \leq i \leq k-1 \\ k+q+1 \leq i \leq n-1 \end{matrix}$$

Also,

$$t_{k+q+1} - t_k = \frac{(q+1)}{n}$$

It will again be assumed that the end conditions $y_0 = y_n = 0$ are to be satisfied, and that the energy constraint (2) holds.

It can easily be seen that the bandwidth defined by (14) can be expressed in the same form as (8), where now the A matrix must be modified to account for the samples removed. The changes, however, are limited to rows and columns of index $k, k+1, k+2, \dots, k+q+1$. The changes in A' are shown in the submatrix below:

$$\begin{matrix} & \begin{matrix} k\text{th column} \\ \text{of } A' \end{matrix} & & \begin{matrix} (k+q+1)\text{th column} \\ \text{of } A' \end{matrix} \\ & \downarrow & & \downarrow \\ \begin{matrix} k\text{th row} \\ \text{of } A' \end{matrix} \longrightarrow & \begin{bmatrix} 1 - \frac{1}{(q+1)^2} & 0 & 0 & \dots & \frac{1}{(q+1)^2} \\ 0 & 2 & 0 & \dots & 0 \\ 0 & 0 & 2 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ \vdots & & & & \vdots \end{bmatrix} & & \\ \begin{matrix} (k+q+1)\text{th row} \\ \text{of } A' \end{matrix} \longrightarrow & \begin{bmatrix} \frac{1}{(q+1)^2} & & & & 1 - \frac{1}{(q+1)^2} \end{bmatrix} & & \end{matrix}$$

The design again reduces to an eigenvalue problem at this point where the matrix in this case is the modified version of A' . The determination of the eigenvalues for this matrix is a more difficult task than for the discrete signal design problem with no samples removed. We cannot solve

for the λ 's in explicit form as before for the noise free case, but a closed form relationship has been derived from which the desired eigenvalues can be determined given numerical values for n , k , and q . This relationship as well as those for the corresponding vector Y , will be omitted due to lack of space.

To conform to a practical situation, the burst location k is assumed to be uniformly distributed over the normalized time interval. For the discrete case this implies that all discrete values of k from 0 to $n-q-1$ are equally likely. It should be noted, however, that the results can be applied to any assumed distribution of burst location. The amplitude averaged discrete signals, for a particular n and q , are then of the form

$$y_i(n, q) = \sum_{k=0}^{n-q-1} y_i(n, q, k) \quad (i = 1, 2, \dots, n-1)$$

The composite vector $Y = (y_1, y_2, \dots, y_{n-1})$ must be re-normalized with respect to unit energy. The y_i 's used here and subsequently are understood to be functions of n and q only.

A discrete signalling alphabet for a particular sample size and burst width is made up of the normalized amplitude averaged Y vectors corresponding to all the eigenvalues of the modified version of A' . As a result of the matrix transformation of (12) all of the non-degenerate eigenvalues are in the range $-2 < \lambda < 2$. It can be shown that the degenerate eigenvalue $\lambda = 2$ is of multiplicity q . Since the matrix whose eigenvalues we are interested in is of order $n-1$ the derived signal alphabet is of order $n-q-1$. This points out a limitation of this design procedure for large burst widths since for a fixed sample size the order of the alphabet decreases as the burst width increases. In addition, it can be shown that the set of eigenvectors corresponding to the distinct eigenvalues for a particular burst location and width form an orthogonal set. However, amplitude averaging over all burst locations does not preserve the orthogonality properties of the alphabets. Therefore, the "useful" alphabet will consist of fewer than $n-q=1$ signals in general.

With the aid of a digital computer program averaged signal alphabets have been constructed for two sample sizes, $n=7$ and $n=13$. For each sample size a "narrow" burst and a "wide" burst case was considered, yielding a total of four signalling alphabets. It can be shown that in the presence of gaussian noise and burst noise with a uniformly distributed burst location, the overall probability of error depends on the various

vector inner products. The "distance" between a pair of signals Y_i and Y_j is given by $D_{ij} = \frac{1}{2} Y_i Y_j^T$. Discrete bandwidths and pairwise distance functions have been calculated for the four alphabets discussed.

For minimum probability of error it is desirable that the absolute values of the distances be as low as possible. For the purpose of specifying useful alphabets we will arbitrarily assume that a pair of signals may be part of the same alphabet provided the absolute value of the distance function for the pair does not exceed 0.20. This allows for the negatives of the given signals to be used as well. The largest alphabets that can be formed for the four examples are given in the table below, subject to the above mentioned distance function threshold. The maximum and r.m.s. bandwidths of the alphabets are also shown, where the bandwidths⁵ have been normalized with respect to that of a half cycle sine pulse with unit energy.

n	q	Alphabet Size ⁺	$B_{\max}$	B_{rms}
7	1	3 signals	4.01	2.87
7	3	2 signals	3.98	3.08
13	2	5 signals	8.00	5.54
13	6	4 signals	7.98	5.63

It is interesting to note that for the case $n=13$ the total alphabet consists of ten signals for the narrow burst case ($q=1$) and six signals for the wide burst case ($q=6$); however, the useful alphabets comprise five and four signals respectively. Wide bursts are therefore not as detrimental to signalling alphabet size as one might expect from the fact that total alphabet size is limited to $n-q-1$ signals. Useful alphabet sizes differ by only one signal for narrow and wide burst cases ($n=13$) whereas total alphabet size differs by four signals. The same trend is seen for the case $n = 7$. Also, burst width has no appreciable affect on alphabet bandwidth as observed from the results. This is true whether the maximum bandwidth or r.m.s. bandwidth definition is used.

IV. DISCRETE SIGNAL DESIGN USING TWO-PLAYER DIFFERENTIAL GAME

A distinctly different approach to the signal design problem is now presented, treating it as a "two-player" differential game. The introduction of a second player is accomplished mathematically by increasing the number of state variables by one and by using the initial value of the new state variable as a design parameter. This results in flexibility of design. Also, a modified version of the Gabor bandwidth criterion is used as the payoff; the effect of this modification on the continuous version of the problem is to soften the penalty to signals having steep slopes near the edges of the interval. By judiciously choosing the value of this parameter we obtain a signal which spreads energy toward the edges of the time interval to the degree desired. In this case the design parameter is analogous to the burst width parameter q used previously.

The following kinematic equations, payoff, and boundary conditions are chosen for the two player game:

+ excluding the negative waveforms

$$\dot{t} = 1, \dot{y} = \phi, \dot{w} = \psi \quad (-1 \leq \psi \leq 1), \dot{u} = y^2, G = \phi^2 w.$$

$$t_0 = 0, t_f = \frac{1}{2}, y(t) = y(1-t) \text{ for } \frac{1}{2} \leq t \leq 1.$$

The details of the solution of the continuous version of this two-player game are given in a previous paper.⁴ It can be shown that in the discrete case a difference equation in y results, and the discrete amplitudes satisfying this equation are given by

$$y_0 = a, y_1 = a, y_2 = \delta, y_3 = 2(\delta + as)$$

$$y_i = 2y_{i-1} - (1 - \frac{s}{k-2})y_{i-2} \quad (4 \leq i \leq k)$$

where the constants a and δ are evaluated from the energy constraint and end condition $y_0 = 0$. The parameter $S = w_k$ allows design flexibility.

For the case $k = 6$ it has been found that the discrete signals for the range $-1.47 < s < -0.4$ are analogous to the minimum bandwidth amplitude averaged signals that are obtained using the one player approach. These signals are characterized by the same properties as for the amplitude averaged signals, namely, positive discrete values over the interval and the spreading of energy to the sides of the interval. The results indicate that, in general, a suitable four signal alphabet can be formed using a signal corresponding to a particular s and another corresponding to most values of s having opposite sign. A six signal alphabet is possible if the vectors corresponding to $s = 5.00, 0.00, -5.00$ and their respective negatives are used. The distance function for the pair corresponding to $s = 5.00$ and $s = 0.00$ is 0.26, only slightly above the arbitrarily chosen threshold of 0.2. It must be remembered that the resulting alphabet, whether four or six signals, is not necessarily optimum in the r.m.s. or maximum Gabor bandwidth sense. However, judicious selection can result in favorable trade-off between bandwidth and detectability in a burst noise environment. It is felt that some advantage may be gained by using more samples. Until further investigation is performed, however, it must be concluded that the two-player game approach can only yield low dimension alphabets.

REFERENCES

1. D. Gabor, "Theory of Communication," J. IEE (London), Vol. 93, (1946).
2. D. Tufts, D. Shnidman, "Optimum Waveforms Subject to both Energy and Peak-Value Constraints," Proc. IEEE, Vol. 52, (1964).
3. D. Ditoro, K. Steiglitz, "Application of Maximum Principle to the Design of Minimum Bandwidth Pulses," IEEE Transactions on Communication Technology, Vol. Com-13, No.4, (1965).
4. H. Silver, L. Kurz, "A Signal Design Problem in Burst Noise using Differential Game Techniques, Proceedings of the Princeton Conference on Information Sciences and Systems, (1968).
5. A. Nuttall, F. Amoroso, "Minimum Gabor Bandwidth of M. Orthogonal Signals," IEEE Transactions on Information Theory, Vol. IT-11, No.3, (1965).
6. M. Simon, L. Kurz, "Optimal Processing and Design of Digital Signals Perturbed by Gaussian and Nongaussian Noise," Tech. Report 400-121, New York University, (1965).
7. R. Isaacs, Differential Games, John Wiley, (1965).

A THEOREM CONCERNING THE DC EQUATIONS OF NONLINEAR NETWORKS

I. W. SANDBERG and A. N. WILLSON, JR.
Bell Telephone Laboratories, Incorporated
Murray Hill, New Jersey

ABSTRACT

We consider the equation

$$F(x) + Ax = B \quad (1)$$

(with $F(\cdot)$ a "diagonal" nonlinear mapping of real Euclidean n -space E^n into itself, and A a real $n \times n$ matrix) which plays a central role in the d-c analysis of transistor networks, and give necessary and sufficient conditions on A such that the equation possesses a unique solution x for each real n -vector B and each strictly monotone increasing $F(\cdot)$ that maps E^n onto itself.

SUMMARY*

For each positive integer n let $\mathcal{F}^n$ denote that collection of mappings of the real n -dimensional Euclidean space E^n onto itself defined by:

$F \in \mathcal{F}^n$ if and only if there exist, for $i = 1, \dots, n$, strictly monotone increasing functions f_i mapping E^1 onto E^1 such that, for each $x \equiv (x_1, \dots, x_n)^t \in E^n$, $F(x) = (f_1(x_1), \dots, f_n(x_n))^t$.

The following notation will also be used: The origin in E^n will be denoted by θ . If D is a diagonal matrix then $D > 0$ ($D \geq 0$) means that each element of D on the main diagonal is positive (nonnegative).

The class of matrices denoted by P_0 has been defined by M. Fiedler and V. Pták in [3]. They in fact prove that the following properties of a square matrix A are equivalent:

- 1) All principal minors of A are nonnegative.
- 2) For each vector $x \neq \theta$ there exists an index k such that $x_k \neq 0$ and $x_k y_k \geq 0$ where $y = Ax$.
- 3) For each vector $x \neq \theta$ there exists a diagonal matrix $D_x \geq 0$ such that $\langle x, D_x x \rangle > 0$ and $\langle Ax, D_x x \rangle \geq 0$.
- 4) Every real eigenvalue of A , as well as of each principal sub-matrix of A , is nonnegative.

* Since the material for this paper is contained in the more comprehensive (but rather long) paper [1] (see also [2]) we include here only a summary of the results.

The class of matrices satisfying one of the above conditions is denoted by P_0 .

The main purpose of this paper is to report on the following result concerning the equation (1).

Theorem. If A is an $n \times n$ matrix then there exists a unique solution of (1) for each $F \in \mathcal{F}^n$ and each $B \in E^n$ if and only if $A \in P_0$.

Equation (1) is encountered often in the study of nonlinear electrical networks. In the case of networks containing only resistors (i.e., linear resistors with nonnegative resistance), dependent and independent sources, and two-terminal nonlinear resistors that are described by functions in $\mathcal{F}^1$ (diodes, for example), this is rather obvious.

More generally, it is frequently the case that networks which contain transistors, as well as the previously mentioned linear and nonlinear elements, may be described by the equation

$$TF(x) + Ax = B. \quad (2)$$

In this case, x is a vector whose components are each of the voltages across the nonlinear resistors and each of the transistor base-emitter and base-collector voltages. The $n \times n$ matrix A is the y -parameter matrix of the linear n -port network which is obtained by removing all nonlinear resistors and transistors and setting the value of each independent source to zero. The function $TF(x)$ describes the behavior of the nonlinear resistors and the transistors. It happens that the matrix T is nonsingular, and therefore (2) can be put into the form of (1).

Networks which contain inductors and capacitors as well as the memoryless elements already mentioned are of course described by differential equations. Even the study of such networks, however, can often lead to the consideration of equations of type (1). One usually finds that the solution of such an equation is necessary, for example, when computing the solution of the differential equations by using some implicit numerical integration formula.

The problem of determining the equilibrium states of the above-mentioned dynamic networks is one in which the consideration of equations of type (1) often arises in perhaps a more direct manner. In this regard, if it happens that (1) has a unique solution, then of course the network cannot possibly be bistable.

REFERENCES

1. I. W. Sandberg and A. N. Willson, Jr., "Some Theorems on Properties of DC Equations of Nonlinear Networks," Bell Sys. Tech. J., January, 1969.
2. I. W. Sandberg and A. N. Willson, Jr., "Some Network-Theoretic Properties of Nonlinear DC Transistor Networks," to appear.
3. M. Fiedler and V. Pták, "Some Generalizations of Positive Definiteness and Monotonicity," Numer. Math., 9, pp. 163-172; 1966.

I. W. SANDBERG and A. N. WILLSON, JR.
 Bell Telephone Laboratories, Incorporated
 Murray Hill, New Jersey

ABSTRACT

We consider several aspects of the problem of determining bounds on the location of the solution of the equation

$$F(x) + Ax = B \quad (1)$$

(with $F(\cdot)$ a "diagonal" nonlinear mapping of real Euclidean n -space E^n into itself, and A a real $n \times n$ matrix) which plays a central role in the d-c analysis of transistor networks. We also show how this problem is related to the problem of determining the existence and uniqueness of a solution of the above equation.

SUMMARY*

We consider several aspects of the problem of determining bounds on the location of the solution of (1). The determination of bounds is important for several reasons. One is, that knowledge of such bounds can often be helpful in the computation of the solution of (1).

Using the same notation as in our other paper in these proceedings [3], we first prove:

Theorem: If $F \in \mathcal{F}^n$, $A \in P_0$, and, for $i = 1, \dots, n$, $\alpha_i \leq \beta_i$ are given, then there exist real numbers $\gamma_i \leq \delta_i$ such that, for any $B \equiv (b_1, \dots, b_n)^t \in E^n$ with $\alpha_i \leq b_i \leq \beta_i$ for $i = 1, \dots, n$, if x satisfies (1) then $\gamma_i \leq x_i \leq \delta_i$ for $i = 1, \dots, n$.

The proof of this theorem shows, in fact, how the numbers γ_i and δ_i can be computed. This result is then used to prove:

Theorem: If A is an $n \times n$ matrix then $A \in P_0$ if and only if for each $F \in \mathcal{F}^n$ and each unbounded sequence of points $x^1, x^2, x^3, \dots$ in E^n , the corresponding sequence $B^1, B^2, B^3, \dots$ ($B^k = F(x^k) + Ax^k$, $k = 1, 2, \dots$) is unbounded.

* Since the material for this paper is contained in the more comprehensive (but rather long) paper [1] (see also [2]) we include here only a summary of the results.

The last theorem, along with the existence and uniqueness theorem proved by the authors in [1] (and stated in [3]), show that: Every bounded input sequence $B^1, B^2, B^3, \dots$ is mapped by (1) into a bounded output sequence $x^1, x^2, x^3, \dots$, for each $F \in \mathcal{F}^n$, if and only if $A \in P_0$.

A matter that is closely related to the proofs of the above theorems on the boundedness of solutions of (1) is that of proving that: For each $F \in \mathcal{F}^n$ and each $A \in P_0$ the solution x of (1) is a continuous function of the vector B .

Finally, using a theorem of R. S. Palais [4] (see also [5]), the relationship between the above results on the boundedness of solutions, and the matter of proving the existence and uniqueness of a solution of (1) is examined.

REFERENCES

1. I. W. Sandberg and A. N. Willson, Jr., "Some Theorems on Properties of DC Equations of Nonlinear Networks," Bell Sys. Tech. J., January, 1969.
2. I. W. Sandberg and A. N. Willson, Jr., "Some Network-Theoretic Properties of Nonlinear DC Transistor Networks," to appear.
3. I. W. Sandberg and A. N. Willson, Jr., "A Theorem Concerning the DC Equations of Nonlinear Networks," these proceedings.
4. R. S. Palais, "Natural Operations on Differential Forms," Trans. Amer. Math. Soc., 92, pp. 125-141; 1959.
5. C. A. Holzmann and R. Liu, "On The Dynamical Equations of Nonlinear Networks with n-Coupled Elements," Proc. Third Ann. Allerton Conf. on Ckt. and System Thy., pp. 536-545; 1965.

CHARACTERISTICS OF A SUBOPTIMUM PHASE-LOCK DEMODULATOR

JOHN J. UHRAN, JR.
University of Notre Dame
Notre Dame, Indiana

ABSTRACT[†]

This paper presents a suboptimum solution to the design of phase-lock loops which yield improved performance in the important low signal-to-noise region. Three loop characteristics of major importance are treated: signal demodulation, lock range and threshold. Experimental curves are presented and compared with previously known results. The approach proves to be simple in form involving only the addition of a specific non-linearity (zero-memory) into the loop's forward path.

INTRODUCTION

The addition of linear filters⁽¹⁾ into a phase-lock loop is well documented as are the effects of VCO and phase detector nonlinearities⁽²⁾ when the input signal-to-noise ratio is high. Despite the fact that non-linear filtering elements do enhance certain system performance indices, only recently has any attempt been made in this direction for phase-lock loops^(3,4,5). This is due in part to the fact that the synthesis problem, the search for an optimum loop design in some sense, is very difficult especially in the low SNR region. Thus, one frequently resorts to ad-hoc approaches based on past experience and results and ultimately tests the hypothesis by appropriate analysis and/or experimentation. In this paper the author uses this method to circumvent the synthesis problem. An experimental study has been made of the effect which a specific non-linearity (zero-memory), $f(e)$, inserted into the forward path of a simple (first order) loop, has on three important loop characteristics: signal demodulation, lock range and threshold in the presence of severe noise.

SYSTEM DESCRIPTION

Fig. 1 represents the system model under consideration. The input $x(t)$,

$$x(t) = \sqrt{2} A \sin(\omega_0 t + \lambda_1(t)) + n(t) \quad (1)$$

[†]Partially supported by NSF GK-1547

is the sum of a fixed amplitude carrier and narrowband noise. For our purposes, $\lambda_i(t)$ is an f.m. modulated signal plus a fixed frequency component, .

$$\lambda_i(t) = \Delta\omega t + m_f \sin \omega_m t \quad (2)$$

where m_f = f.m. modulation index. The input noise has a single-sided spectral density of N_0 . The VCO output signal is represented as

$$y(t) = \sqrt{2} \cos(\omega_0 t + \beta \int_{-\infty}^t e_0(u) du) \quad (3)$$

where β = VCO gain in rad./volt. The signal $e_0(t)$ is the demodulated information signal, in this case sinusoidal, when no noise is present. The bandwidth of the ideal low-pass filter is ω_m in the case of sinusoidal modulation.

The non-linearity, $f(e)$, is described by eq.(4).

$$f(e) = \begin{cases} ae + b & e > C \\ e & |e| \leq C \\ ae - b & e < -C \end{cases} \quad (4)$$

where $0 \leq a \leq 1$

and $b = C(1 - a)$.

When $a=1$, $f(e)$ is linear and we have a simple first order phase-lock loop, while for $a=0$, $f(e)$ becomes a hard limiter. The breakpoint, C , of $f(e)$ is determined by the maximum value of $|e|$ which still allows synchronous operation. In the model of Fig. 1, $C=A$, the rms value of the input signal.

This particular choice of $f(e)$ was predicated on the following basic principles:

- (i) that no effect on the system occurs when noise is absent,
- (ii) that the loop gain at any operating point not exceed that at $f(0)$,
- (iii) that the basic modulo 2π characteristic of the simple loop be preserved,
- (iv) that the non-linearity should improve the lock range and cycle-slipping properties of the system in the presence of noise and,
- (v) that the demodulation properties of the loop be improved.

Inspection of eq.(1),(3) and (4) indicates that (i),(ii) and (iii) are immediately satisfied. By design, $f(e)$ is effectively eliminated from the system model when $n(t)=0$. As a result, the signal only characteristics of this system are identical to those of the well-studied first order loop^(1,6) and are not considered here.

SYSTEM PERFORMANCE

The experimental model used in the laboratory conformed closely to that described by eqs.(1-4). For convenience the equivalent noise bandwidth of the loop, B_L ,⁽¹⁾ was 172.5Hz. The independent variable used throughout is the normalized input signal-to-noise ratio, α , which is defined as

$$\alpha = A^2/N_o B_L \quad (5)$$

Cycle-Slipping Performance

In systems such as this where the system bandwidth is much narrower than the input, spikes or impulses appearing at the output are an excellent indication of cycle-slipping in the loop. The onset of this phenomena thus precipitates threshold. In Fig.2 the average rate of cycle-slipping, $\bar{S}$, is plotted vs. α for several values of a , the parameter of $f(e)$. The frequency difference, $\Delta\omega$, between the input carrier and the VCO reference frequency is zero and no modulation is present, thus giving a "best case" performance. As the degree of saturation increases-decreasing a -cycle-slipping is suppressed. Thus the threshold shows a decided improvement reaching a maximum of about 3.75 db. It should be observed that as $\Delta\omega$ shifts or modulation appears that these threshold curves will shift to the right for each value of a . For $a=1$, the indicated result is close to that predicted by eq.(6)(1).

$$\bar{S}/B_L = 4/\pi e^{-2\alpha} \quad (6)$$

Lock Range

To complement the cycle-slipping characteristics of this system, one must consider the important lock range behavior. When $\Delta\omega$ is no longer zero, such as in tracking applications, threshold does degrade. Fig.3 clearly indicated such performance of the model ($a=0$ and 1) as a function of α . For convenience the results are normalized to the maximum value of lock range ($4B_L$ for a first order loop). The high asymptotes are both one since the no noise performance is identical. As α decreases, the hard limiting case is slightly worse than the simple loop. But in the important low SNR region, the limiter demonstrated a clear advantage, the separation of the two curves being about 3.75 db as before. For convenience $\bar{S}$ was kept constant at 10 spikes/sec.

Output Signal and Noise Performance

When a phase-lock loop is used as an f.m. demodulator, the question of signal-to-noise ratio performance at the output becomes important as does an equivalent model. Fig.4 has been proposed as an equivalent model for the simple phase-lock loop⁽⁷⁾. The noise source, $n_p(t)$, is an impulse generator whose rate is S . The switch in the feedback loop is open during the occurrence of impulses. $n'(t)$ is the low pass equivalent of the input noise $n(t)$. Assuming the main effect of $f(e)$ is impulse suppression, Fig.4 can also be used as an approximate model of Fig.1.

If the input signal, $\lambda_1(t)$, eq.(2) is used with $\Delta\omega=0$, then one obtains the output signal-to-noise ratio as

$$\text{SNR}_o = \frac{3/2 m_f^2 (B_L/f_m) \alpha}{1 + 6(\bar{S}/B_L)(B_L/f_m)^2 \alpha} \quad (7)$$

where it has been assumed that $f_m \ll B_L$ and $f_m m_f \ll 4B_L$. The former permits a simpler expression for SNR_o ; the latter is necessary so that the impulse source is independent of the modulation. Once S/B_L is known (Fig. 2 or eq.(6)) then SNR_o can be found. The right term in the denominator is due to the impulse source.

To test the model and demonstrate actual performance several cases are plotted in Fig.5. A value of $m_f=2$ and $(B_L/f_m)=5$ was chosen as an example. The experimental results are quite close in the high α region. For both cases of a , the model makes a higher prediction than the experiment when α is small. This is due to the fact that the model does not account for the loss of signal energy during a cycle slip⁽⁶⁾.

When $a=0$, the experimental curves indicate a change of slope before cycle-slipping sets in. In this region, due to the peak limiting of the noise, the noise power actually increases more slowly and the signal power increases about +.4 db. Once cycle-slipping sets in a sharp increase of the noise power and a decrease of signal power is observed for both cases. Again the separation of the two curves is approximately 3.75 db below threshold.

CONCLUSIONS

The choice of a specific non-linear filter to improve the low SNR performance of a first-order loop proves to satisfy all the basic criteria as well as being extremely simple to use. The improvements

in the basic performance indices appear to be quite significant. Though actual harmonic distortion of the output signal due to limiting has not been measured, it is not believed to be significant enough to prevent usage of this f(e) in most applications.

REFERENCES

1. A.J.Viterbi, Principles of Coherent Communication, McGraw-Hill, N.Y., 1967.
2. D. Schilling and M.Smirlock, 'Intermodulation Distortion of Phase Locked Loop Demodulator', IEEE Trans on Communication Technology, v.COM-15, n.2, April, 1967.
3. W. Hammond, et. al. 'Design of a Generalized Phase Comparator for Phase-Lock Loops', Proc. 11th MSCT, pp.10-16, May, 1968.
4. J.Uhran, 'Threshold Improvement for Phase-Lock Loop Systems', Electronic Letters, pp.203-204, May 17, 1968.
5. L.Cleland and B.Leon, 'Phased Locked Loops for Large Signal Tracking', Proc. 11th MSCT, pp.2-9, May, 1968.
6. S.Gupta and J.Bayless, 'Threshold Extension Using a Phase Lock Demodulator in a FM Feedback Loop', Proc. National Telemetering Conference, April, 1968.
7. J. Uhran, 'Cycle Slipping Effects on the Output Signal of a Phase-Locked Demodulator', Proc. IEEE, v.56, n.1, pp.80-81, Jan., 1968.

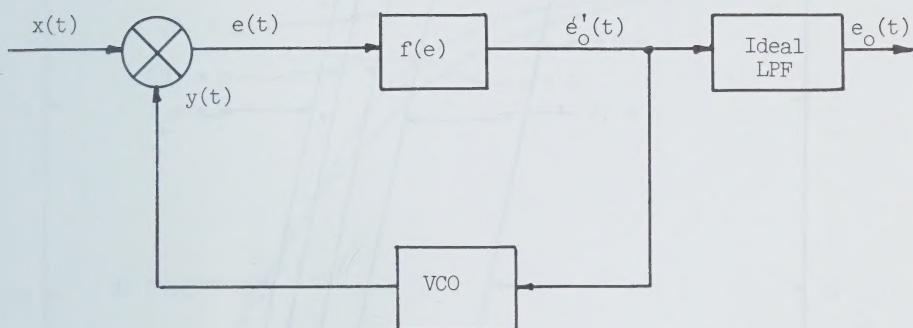


Fig. 1. Simple Phase Lock with a Non-linear Filter.

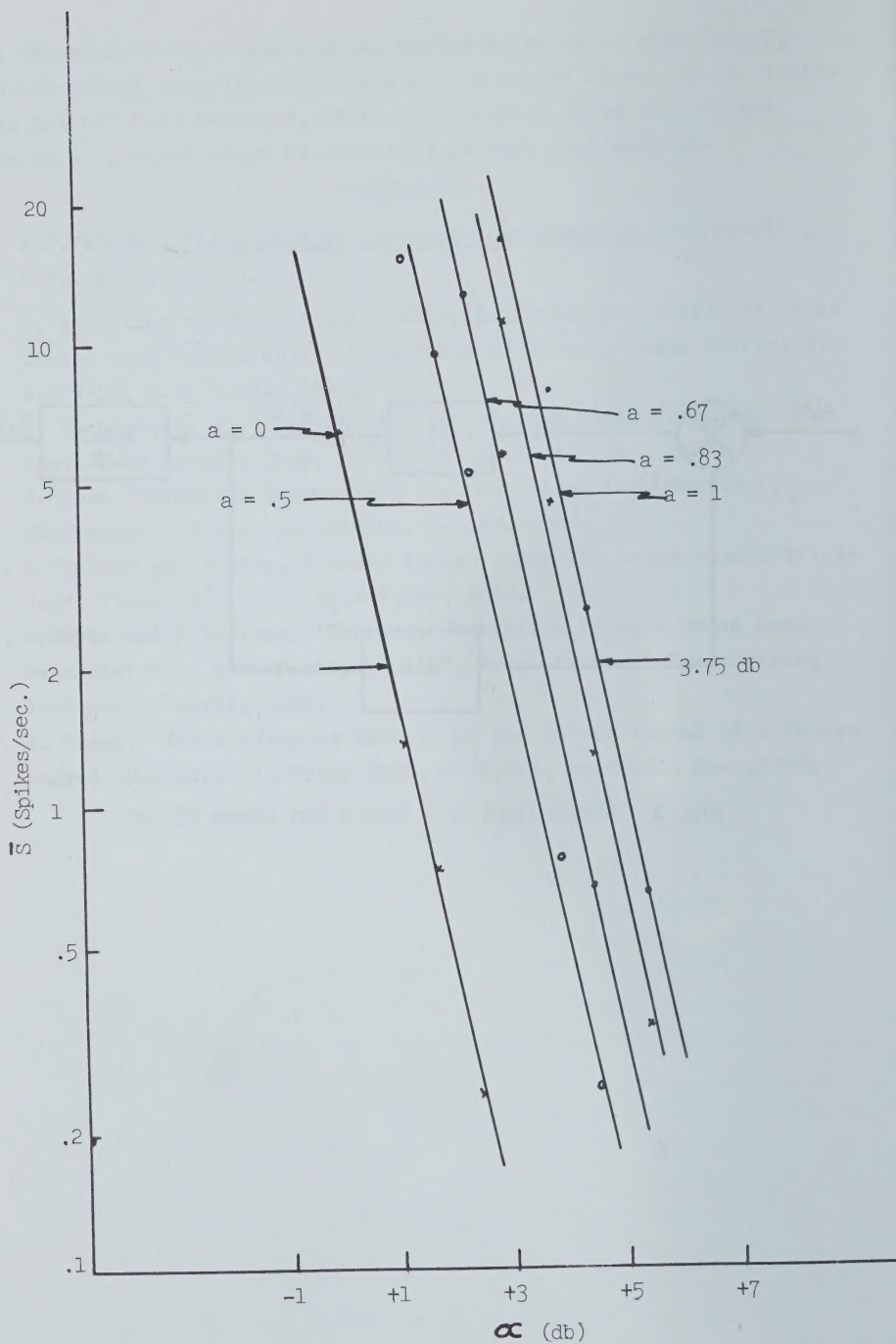


Fig. 2. System Cycle-Slipping Performance.

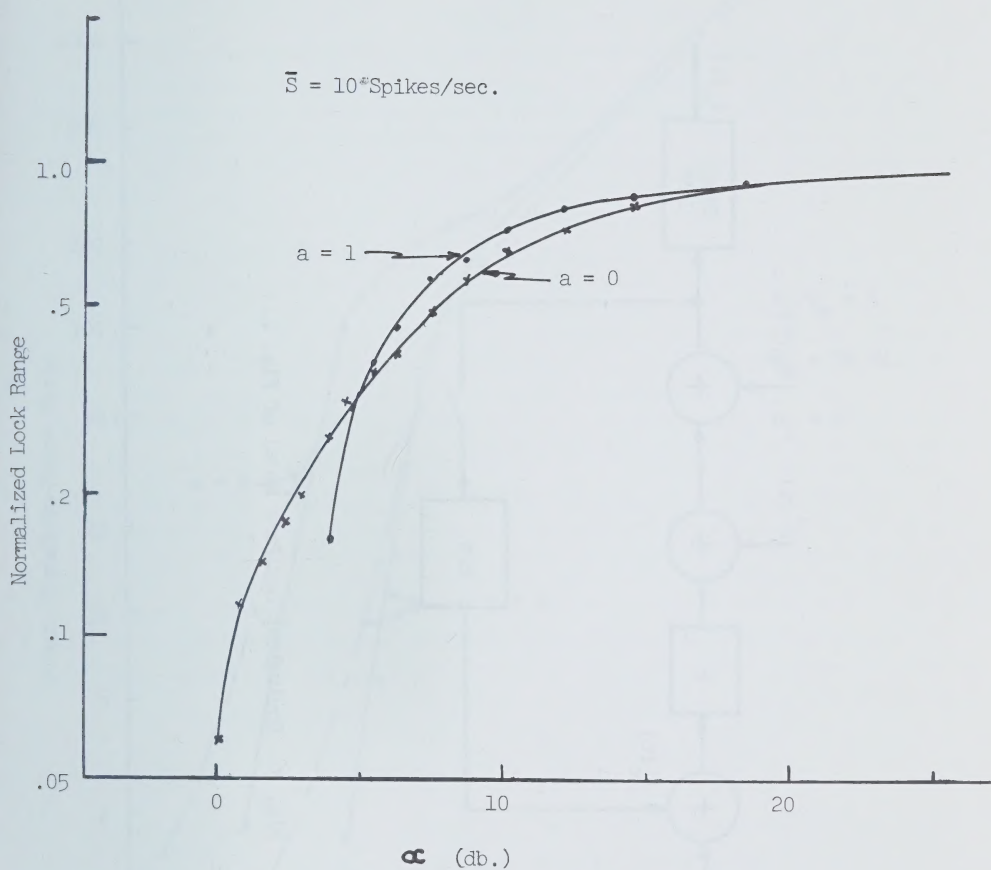


Fig. 3. System Lock-Range Performance.

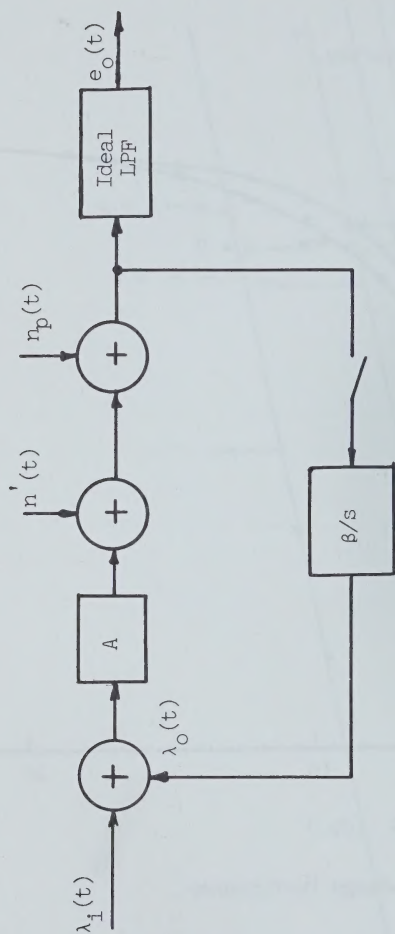


Fig. 4. Equivalent Low Pass Model of Fig. 1.

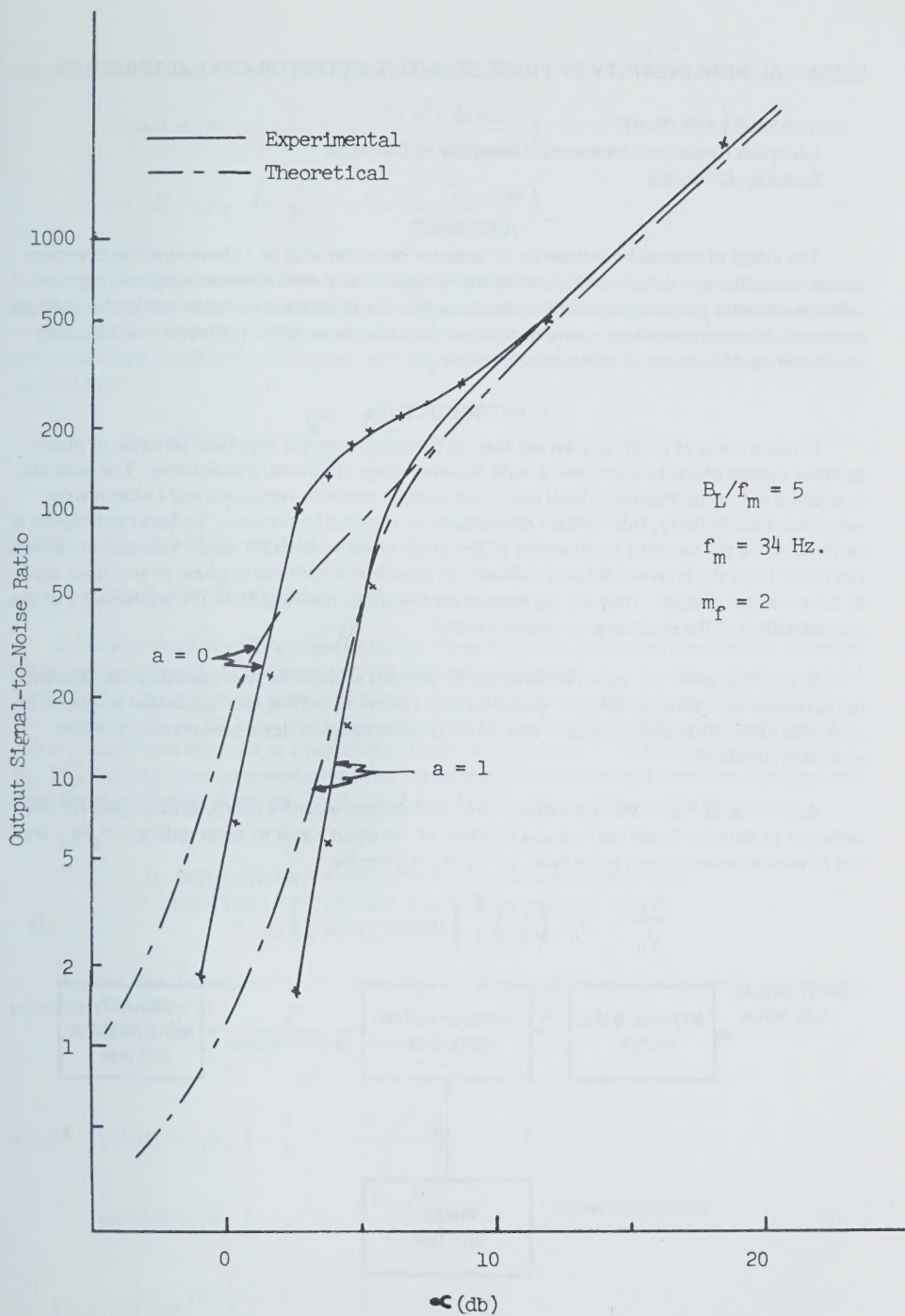


Fig. 5. Output Signal-to-Noise Ratio

ESSENTIAL NONLINEARITY OF PHASE-SENSITIVE DETECTOR CHARACTERISTICS

BRANKO LESKOVAR

Lawrence Radiation Laboratory, University of California
Berkeley, California

ABSTRACT

The effect of essential nonlinearity of detector characteristics in a phase-sensitive detection system is studied and determined theoretically in detail over a wide dynamic range of optimum and nonoptimum operating conditions, assuming that the input signal is in the narrow-band Gaussian noise. Minimum, maximum, and limiting values of nonlinearities of detector characteristics are determined by means of computer-aided analysis.

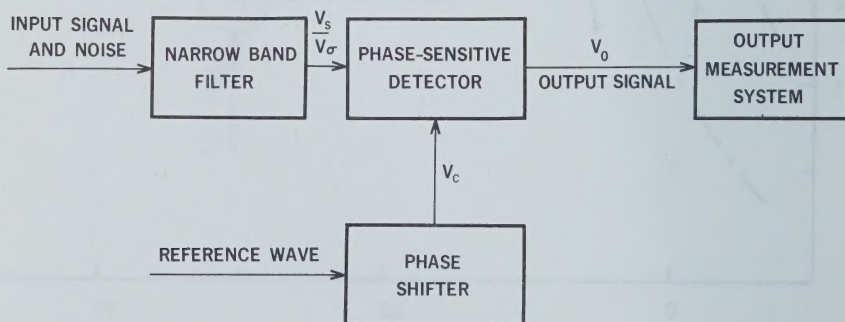
I. INTRODUCTION

In many cases of practical interest, there is concern about the nonlinear behavior of phase-sensitive system characteristics over a wide dynamic range of operating conditions. For example, in wide-band Fourier-transform high resolution nuclear-magnetic-resonance and electron-spin-resonance spectrometry, total system nonlinearity is of prime importance. System nonlinearity is determined by the essential nonlinearity of the phase-sensitive detector used. Essential nonlinearities result from the inherent detector behavior in detection amplitude or phase of the input signal in the presence of noise. They do not involve nonlinearities resulting from the nonlinearity of the characteristics of the electronic components used.

In previous work [1] essential nonlinearity analysis of detector characteristics was presented for particular operating conditions. Based on this, further effort has been expanded by means of computer-aided analysis to include a very wide dynamic range of optimum and nonoptimum operating conditions.

According to Fig. 1, the normalized form of the phase-sensitive characteristics used for determination of detector nonlinearities as a function of the input signal-to-noise ratio $x = V_s/V_\sigma$, and the reference wave-to-input noise ratio $\mu = V_c/V_\sigma$ is given by

$$\frac{V_0}{V_\sigma} = \eta_d \left(\frac{\pi}{2} \right)^{1/2} \left\{ u[v(x)] - y[t(x)] \right\}, \quad (1)$$



XBL 687-1177

Fig. 1. Phase-sensitive detector.

where functions $u[v(x)]$ and $y[t(x)]$ are given by

$$u[v(x)] = {}_1F_1 \left(-\frac{1}{2}; 1; -\frac{\mu^2 + x^2 + 2\mu x \cos \psi}{2} \right), \quad (2)$$

$$y[t(x)] = {}_1F_1 \left(-\frac{1}{2}; 1; -\frac{\mu^2 + x^2 - 2\mu x \cos \psi}{2} \right). \quad (3)$$

By using a notation consistent with Ref. 1, V_0 is the detector output signal, V_σ is the root-mean-square value of the input narrow-band noise, η_d is the detection efficiency, V_s is the amplitude of the input sine signal, V_c is the amplitude of the reference wave, ψ is the phase angle between the input signal and the reference wave, and ${}_1F_1$ denotes the confluent hypergeometric function defined by

$${}_1F_1(a; b; \pm p) = \sum_{n=0}^{\infty} \frac{(a)_n}{(b)_n} \cdot \frac{(\pm p)^n}{n!}, \quad (4)$$

$$\text{where } (a)_n = a(a+1) \dots (a+n-1), \quad (a)_0 = 1, \quad (5)$$

$$(b)_n = b(b+1) \dots (b+n-1), \quad (b)_0 = 1.$$

Generally, three important cases of the nonlinearity of the phase-sensitive detector characteristics have been found. In the first case, the detector nonlinearity N_A has been determined as a function of x for the phase angle ψ between the input signal and the reference wave equal to $2n\pi$, where $n = 0, \pm 1, \pm 2, \dots$. In the second and the third cases, the detector nonlinearities N_B and N_C have been obtained as a function of the phase angle and x . In all cases, nonlinearities N_A , N_B , and N_C have been evaluated from the extent of their departure from the tangent drawn from the characteristic at points $x = 0$, $\psi_2 = (2n+1)\pi/2$, and $\psi_1 = 2n\pi$, respectively.

II. NONLINEARITY OF DETECTOR CHARACTERISTICS RELATING TO THE INPUT SIGNAL-TO-NOISE RATIO FOR PHASE ANGLE $\psi = 2n\pi$

For this case the essential nonlinearity N_A can be calculated over a wide range of operating conditions by means of

$$N_A = 1 - \frac{1}{x\mu} \frac{\gamma[f(x)] - \omega[\epsilon(x)]}{\chi(\mu)}, \quad (6)$$

$$\text{where } \gamma[f(x)] = {}_1F_1 \left(-\frac{1}{2}; 1; -\frac{\mu^2 + x^2}{2} \right), \quad (7)$$

$$\omega[\epsilon(x)] = {}_1F_1 \left(-\frac{1}{2}; 1; -\frac{\mu^2 - x^2}{2} \right), \quad (8)$$

$$\chi(\mu) = {}_1F_1 \left(\frac{1}{2}; 2; -\frac{\mu^2}{2} \right) \quad (9)$$

By means of a digital computer, the nonlinearity N_A as a function of x with μ as parameter is calculated and plotted (Fig. 2). High-accuracy numerical calculations of the confluent hypergeometric function in Eq. (6) are performed in two different ways. Where the variable of the hypergeometric function is smaller than 20, the function is expressed and calculated in terms of the modified Bessel functions of the first kind according to the relations²

$${}_1F_1\left(-\frac{1}{2}; 1; -p\right) = \exp\left(-\frac{p}{2}\right) \left[(1+p) I_0\left(\frac{p}{2}\right) + x I_1\left(\frac{p}{2}\right) \right] , \quad (10)$$

$${}_1F_1\left(\frac{1}{2}; 2; -p\right) = \exp\left(-\frac{p}{2}\right) \left[I_0\left(\frac{p}{2}\right) + I_1\left(\frac{p}{2}\right) \right] . \quad (11)$$

For the hypergeometric function variable larger than 20, asymptotic expansion is used according to the following formula²

$${}_1F_1(a; b; -p) \sim \frac{\Gamma(b)}{\Gamma(b-a)} p^{-a} \sum_{n=0}^{\infty} \frac{(a)_n (a-b+1)_n}{n! p^n} , \quad (12)$$

where $\text{Re } p > 0$, $(a)_n$ is defined by Eq. (5), and

$$(a-b+1)_n = (1+a-b)(2+a-b) \dots (n+a-b); (a-b+1)_0 = 1 . \quad (13)$$

The total error of the calculated nonlinearity values $N_A \approx 5 \times 10^{-7}$ if one uses expression (4) and the first eight terms on the right-hand side of Eq. (12), including computer limitations.

Curves in Fig. 2 show that the nonlinearity N_A can be considerably decreased by increasing μ/x . For example, increasing μ/x for one order of magnitude will decrease the nonlinearity by three orders of magnitude, for $x = 1$. Furthermore, the nonlinearity is even more reduced for

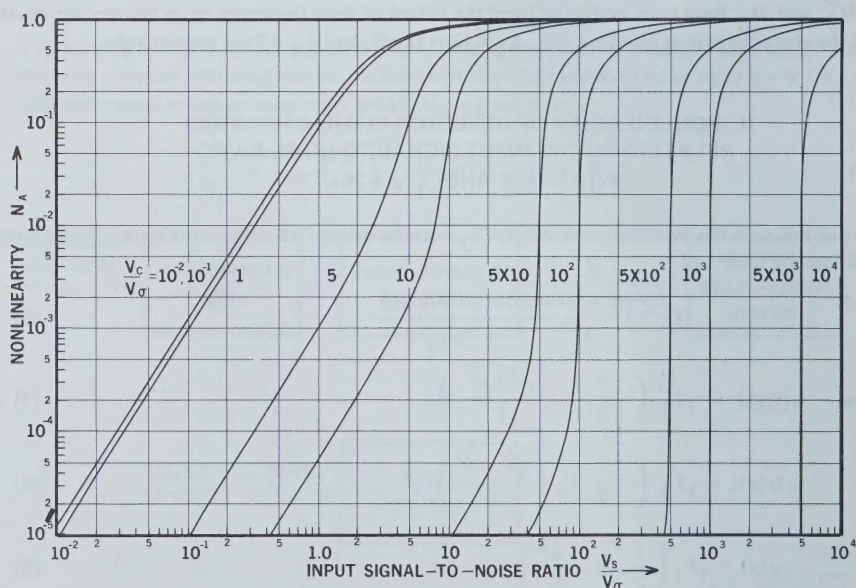


Fig. 2. Behavior of the nonlinearity N_A for the phase angle $\psi = 2n\pi$.

XBL 687-1178

larger amounts of x . It can be seen that for $\mu = 10$ increasing μ by one order of magnitude decreases the nonlinearity more than four orders of magnitude. Consequently, the gain in the decrease of nonlinearity will be larger for larger x for the same increase of μ/x . However, for purposes of obtaining the largest output signal and a wide dynamic range in the wide-band phase-sensitive detection application by using solid-state components, x is close in value to μ . In such applications the essential nonlinearity can have an appreciable value, especially for small μ and x . With increasing μ and x the essential nonlinearity is gradually decreased, as can be concluded by a comparison of the curves given in Fig. 2. For $\mu = x$, approximation values of the nonlinearity N_A can be seen from Fig. 2: 8.5, 5.2, 0.51, and 0.035% for x equals 1, 10, 10^2 , and 10^3 , respectively. However, if both μ and x are greater than 10, μ always has to be slightly larger than x . If this condition is not satisfied the nonlinearity can reach an unacceptably high value; e.g., for $\mu = 100$ and $x = 80$ the nonlinearity $\approx 0.01\%$. On the other hand, for $\mu = 100$ and $x = 120$, nonlinearity is more than three orders of magnitude larger, more than approximately 10%. This amount is unacceptable for most practical applications. Generally, the nonlinearity deterioration will be larger for larger values of both μ and x under previously mentioned conditions. Consequently, wherever the essential nonlinearity is of prime importance, the maximum x value should be always smaller than μ , irrespective of other demands that can be imposed on the phase-sensitive detection system for other reasons. This is particularly important in applications in which x varies in a wide dynamic range. If x is only 10% smaller than μ the nonlinearity N_A can be reduced by more than 1.5 orders of magnitude; this can be concluded by inspection and comparison of curves (in Fig. 2) for $\mu = 10^2, 10^3$, and 10^4 .

III. NONLINEARITIES OF DETECTOR CHARACTERISTICS RELATING TO THE PHASE ANGLE BETWEEN THE INPUT SIGNAL AND THE REFERENCE WAVE

The essential nonlinearities N_B and N_C of the detector characteristics with respect to the phase angle between the input signal and the reference wave can be determined over a wide range of operating conditions (according to Ref. 1) by expressions

$$N_B = 1 - \frac{1}{x\mu} \frac{u[v(x)] - y[t(x)]}{w[f(x)] \left(\frac{\pi}{2} - \psi \right)} \quad (14)$$

and

$$N_C = 1 - \frac{u[v(x)] - y[t(x)]}{\phi[g(x)] - z[h(x)]} \quad (15)$$

where other functions are defined by relations (2) and (3) and by

$$w[f(x)] = {}_1F_1 \left(\frac{1}{2}; 2; -\frac{\mu^2 + x^2}{2} \right), \quad (16)$$

$$\phi[g(x)] = {}_1F_1 \left[\frac{1}{2}; 1; -\frac{(\mu + x)^2}{2} \right], \quad (17)$$

$$z[h(x)] = {}_1F_1 \left[\frac{1}{2}; 1; -\frac{(\mu - x)^2}{2} \right]. \quad (18)$$

Nonlinearities N_B and N_C have been calculated in relation to the phase characteristic operating points $\psi_2 = (2n+1)\pi/2$ and $\psi_1 = 2n\pi$, respectively. Both nonlinearities depend upon ψ , x , and μ . By using Eqs. (14) through (18), nonlinearities N_B and N_C are numerically

calculated, and shown in Fig. 3. The signal-to-noise and the reference wave-to-noise ratios of 1, 10^2 , 10^3 , and 10^4 are chosen as parameters.

The curves in the figure show that for $\mu = x = 1$, the nonlinearity N_B will be smaller than 1% if the phase-angle deviation is $\Delta\psi \leq 0.95 \pi/12$. The phase-angle deviation can be $\Delta\psi \leq 1.76 \pi/12$ for the same nonlinearity value if $\mu = x \geq 10$. The maximum nonlinearity $N_{B,\max}$ appears at the point $\psi_1 = 2n\pi$, but its value depends on x and μ . For the case $\mu = x$, the following maximum nonlinearities $N_{B,\max}$ can be calculated from Eq. (14): 35.41, 15.28, 10.52, 10.02, and 9.97% for x of 1, 10, 10^2 , 10^3 , and 10^4 . Generally, the maximum nonlinearity will be smaller for a larger μ and x . However, to obtain a better insight into nonlinearity behavior with x variation, the nonlinearity N_B is calculated and plotted in Figs. 4 and 5 with fixed values of $\mu = 1, 10^2, 10^3$, and 10^4 , and $\psi = 0, \pi/6, \pi/4, \pi/3, 5\pi/12, 11\pi/24, 47\pi/96$, and $99\pi/200$.

The minimum nonlinearity $N_{B,\min}$ is obtained with respect to x , for μ and ψ as parameters, if the following condition is satisfied:

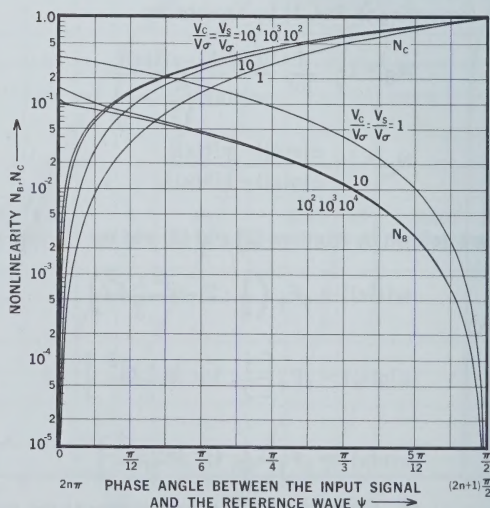
$$w[f(x)] \left\{ \left(\frac{x^2}{2} + \frac{\mu \cos \psi}{2} x \right) s[v(x)] - \left(\frac{x^2}{2} - \frac{\mu \cos \psi}{2} x \right) m[t(x)] + y[t(x)] - u[v(x)] \right\} - \left(\frac{x}{2} \right)^2 K[f(x)] \{ y[t(x)] - u[v(x)] \} = 0, \quad (19)$$

where $w[f(x)]$, $y[t(x)]$, $u[v(x)]$ are given by relations (16), (3), and (2), respectively. The other functions are defined by

$$s[v(s)] = {}_1F_1 \left(\frac{1}{2}; 2; -\frac{\mu^2 + x^2 + 2\mu x \cos \psi}{2} \right), \quad (20)$$

$$m[t(x)] = {}_1F_1 \left(\frac{1}{2}; 2; -\frac{\mu^2 + x^2 - 2\mu x \cos \psi}{2} \right), \quad (21)$$

$$K[f(x)] = {}_1F_1 \left(\frac{3}{2}; 3; -\frac{\mu^2 + x^2}{2} \right). \quad (22)$$



XBL 687-1179

Fig. 3. Comparison of nonlinearities N_B and N_C .

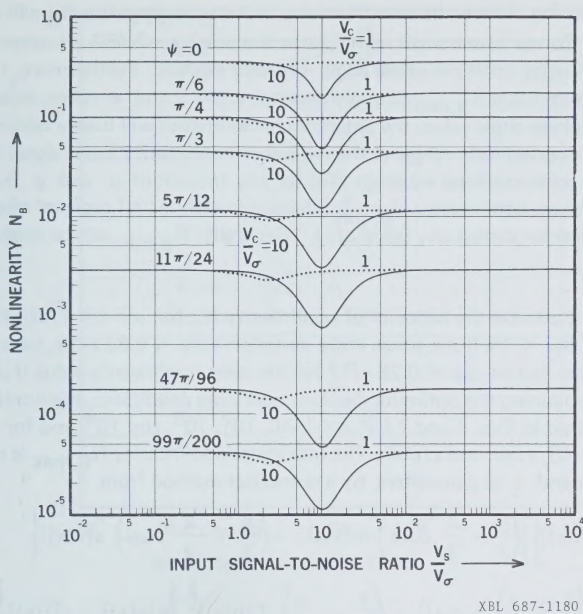


Fig. 4. Nonlinearity N_B as a function of the input signal-to-noise ratio with the phase angle ψ as parameter and the reference wave-to-input noise ratio $V_c/V_\sigma = 1, 10$.

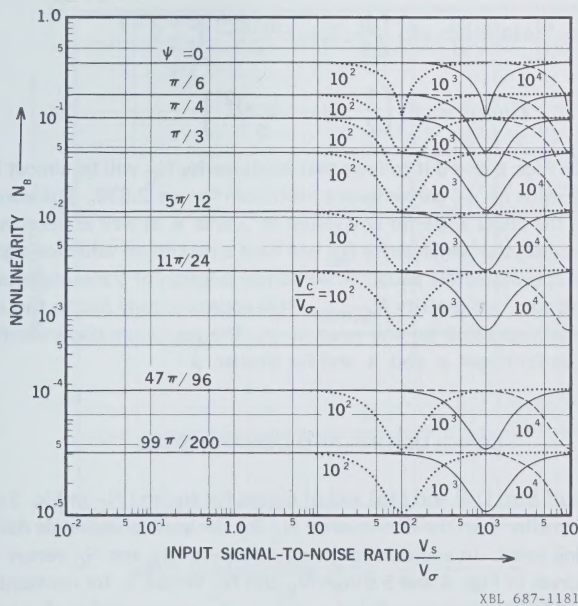


Fig. 5. Nonlinearity N_B as a function of the input signal-to-noise ratio with the phase angle ψ as parameter and the reference wave-to-input noise ratio $V_c/V_\sigma = 10^2, 10^3, 10^4$.

From curves in Fig. 4 it can be seen that for $\mu = 1$ the nonlinearity N_B will be practically independent of x . Curves have a small minimum at the point $x = 2.482$. However, the strong dependence of nonlinearity upon the phase angle is readily evident. Furthermore, for $\mu \geq 10$ the nonlinearity minimum value $N_{B,\min}$ strongly depends upon μ and x ratios, as well as on phase angle ψ . For any phase-angle value, the detector characteristics will have a nonlinearity minimum if μ is close to x . Approximate values of the minimum nonlinearity input signal-to-noise ratio can be found from the transcendental equation (19) for any amount of μ and ψ by a numerical method. The minimum nonlinearity $N_{B,\min}$ is approximately a half order of magnitude smaller than its average value for almost any value of ψ . Generally, $N_{B,\min}$ will be smaller for larger x and μ .

Figure 3 also illustrates the behavior of nonlinearity N_C for $\mu = x = 1, 10, 10^2, 10^3$, and 10^4 . Nonlinearity $N_C < 1\%$ if the phase angle deviation $\Delta\psi \leq 0.53\pi/12$, for $x = \mu = 1$. The phase-angle deviation can be $\Delta\psi \leq 0.28\pi/12$ for the same nonlinearity value if $\mu = x \geq 10$. Furthermore, for obtaining the optimum detector operating conditions, the nonlinearity N_C is calculated and plotted in Figs. 6 and 7 for $\mu = 1, 10, 10^2, 10^3$, and 10^4 ; and for $\psi = 47\pi/96, \pi/3, \pi/4, \pi/12, \pi/24, \pi/48, \pi/96$, and $\pi/300$. The maximum nonlinearity $N_{C,\max}$ is calculated with respect to x , for μ and ψ as parameters, by a numerical method from

$$\left\{ \phi[g(x)] - z[h(x)] \right\} \left\{ \left(\frac{x}{2} - \frac{\mu}{2} \cos\psi \right) m[t(x)] - \left(\frac{x}{2} + \frac{\mu}{2} \cos\psi \right) s[v(x)] \right\} + \left\{ \left(\frac{x}{2} + \frac{\mu}{2} \right) \rho[g(x)] - \left(\frac{x}{2} - \frac{\mu}{2} \right) \ell[h(x)] \right\} \left\{ u[v(x)] - y[t(x)] \right\} = 0, \quad (23)$$

where $u[v(x)]$, $y[t(x)]$, $\phi[g(x)]$, $z[h(x)]$, $s[v(x)]$, and $m[t(x)]$ are given by relations (2), (3), (17), (18), (20), and (21), respectively. Other functions are defined by

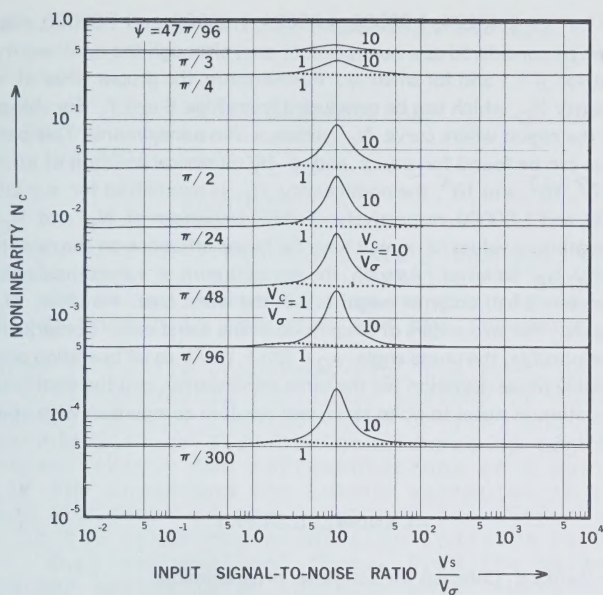
$$\rho[g(x)] = {}_1F_1 \left[\frac{1}{2}; 2; -\frac{(\mu+x)^2}{2} \right], \quad (24)$$

$$\ell[h(x)] = {}_1F_1 \left[\frac{1}{2}; 2; -\frac{(\mu-x)^2}{2} \right]. \quad (25)$$

From curves in Figs. 5 and 6 it follows that nonlinearity N_C will be almost independent of x for $\mu = 1$. For any value of ψ , curves have a maximum for $x = 2.536$. The nonlinearity will strongly depend on the phase angle for any values of μ and x as well as upon x if $\mu \geq 10$. Generally, in these cases, the nonlinearity N_C will have a maximum value for a given ψ , if the ratios μ and x are approximately equal. A numerical solution of transcendental equation (23) gives x of the maximum nonlinearity $N_{C,\max}$. It is approximately one to two orders of magnitude larger than its average value for any value of ψ . The maximum nonlinearity will be larger than its average value for larger μ and x and for smaller ψ .

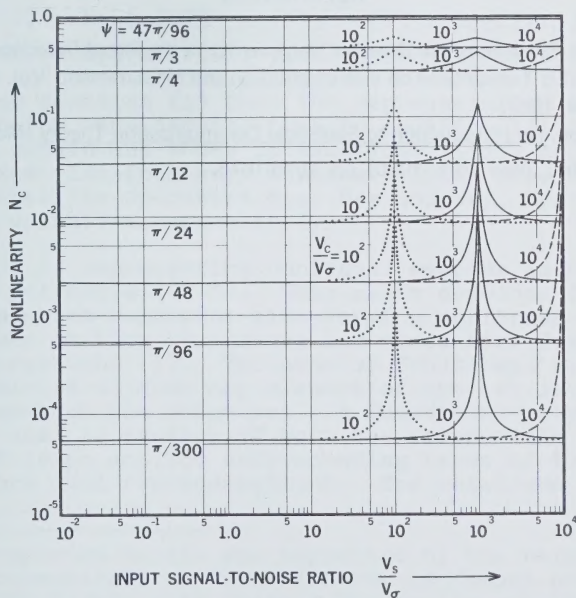
CONCLUSIONS AND COMPARISONS

A comparison of Eqs. (14) and (15) and of curves for N_B and N_C in Fig. 3 shows that the nonlinearity N_B is smaller than the nonlinearity N_C for the same phase-angle deviation $\Delta\psi$ around the operating point. In addition to the above curves (N_B and N_C versus ψ for constant x), another set of curves in Figs. 4 and 5 shows N_B and N_C versus x for constant μ and ψ . The curves of Figs. 4 and 5 show that for a fixed μ and ψ , the proper choice of x will minimize the nonlinearity N_B . The best value of the input signal-to-noise ratio can be found by solution of transcendental equation (19). For $\psi = \pi/4$ and $\mu = 1, 10, 10^2, 10^3$, and 10^4 , the nonlinearity



XBL 687-1182

Fig. 6. Nonlinearity N_C as a function of the input signal-to-noise ratio with the phase angle ψ as parameter and the reference wave-to-input noise ratio $V_c/V_\sigma = 1, 10$.



XBL 687-1183

Fig. 7. Nonlinearity N_C as a function of the input signal-to-noise ratio with the phase angle ψ as parameter and the reference wave-to-input noise ratio $V_c/V_\sigma = 10^2, 10^3, 10^4$.

N_B is minimized for $x = 2.48274, 1.00935, 1.00004, 1.00000$, and 1.00000 , respectively. Any other x value can be considered as a nonoptimum one, although the nonlinearity increase will not be significant for $\mu = 1$ and for small ψ . Furthermore, the proper value of x will also minimize the nonlinearity N_C , which can be concluded from Figs. 6 and 7. For this purpose it is important to avoid the region where curve N_C versus x has a maximum. This particularly nonoptimum x value can be found for any x and ψ by numerical solution of Eq. (23). If $\psi = \pi/24$ and $\mu = 1, 10, 10^2, 10^3$, and 10^4 , the nonlinearity N_C is maximized for $x = 2.53699, 1.01915, 1.00010, 1.00000$, and 1.00000 , respectively. From comparison of N_B and N_C curves it can be seen that nonoptimum values of x will have far larger influence on the nonlinearity N_C than on the nonlinearity N_B behavior. As such, the nonoptimum x values maximally increase the N_B for approximately a half order of magnitude in the worst case. However, N_C nonoptimum x values increase N_C for two orders of magnitude in the worst case. Consequently, one should choose, whenever possible, the phase angle $\psi_2 = (2n + 1)\pi/2$ as an operating point. In this case the larger permissible phase deviation for the same nonlinearity, and the significantly smaller nonlinearity for nonoptimum signal-to-noise ratio, will result in comparison with operating point $\psi = 2n\pi$ (case of N_C).

ACKNOWLEDGMENT

The author thanks E. Coleman for computer programming.

This work was done under the auspices of the U. S. Atomic Energy Commission.

REFERENCES

1. B. Leskovar, Phase-Sensitive Detector Nonlinearity at the Signal Detection in the Presence of Noise, IEEE Transactions on Instrumentation and Measurement, Vol. 1M-16, No. 4, pp. 285-294, 1967.
2. D. Middleton, An Introduction to Statistical Communication Theory (McGraw-Hill Book Company, Inc., New York, 1960), pp. 1075-1076.

CHARACTERIZATION OF TIME-VARYING NONLINEAR SYSTEMS

M. AROZULLAH

Department of Electrical Engineering
Queen's University at Kingston
Kingston, Ontario, Canada.

ABSTRACT

The concept of time-frequency duality has been rather rigorously dealt with and used by P.A. Bello^(1,2) and Allen Gersho⁽³⁾ for characterization of time-varying linear systems in a symmetrical manner in the time and the frequency domains.

This paper is partly concerned with the problem of characterization of time-varying nonlinear systems by dual kernel system functions. A time function $x(t)$ or $y(t)$ and the corresponding spectrum $X(f)$ or $Y(f)$ are considered to be dual variables respectively. Two representations of a system are called dual if the input and the output variables in one representation are duals of the input and the output variables respectively in the other representation and these representations have dual mathematical forms. From the mathematical point of view any system is an operator which operates on its input to produce an output. Both the input and the output of a system can be represented in the time-domain or in the frequency domain. Hence any system can be represented in any one of the following four ways:

$$\begin{array}{ll} (i) & y(t) = H_{tt}[x(t)] \\ (iii) & y(t) = H_{tf}[X(f)] \end{array} \quad \begin{array}{ll} (ii) & Y(f) = H_{ff}[X(f)] \\ (iv) & Y(f) = H_{ft}[x(t)] \end{array} \quad (1)$$

It is seen from equation (1) that the representation in equation 1(i) is in the time domain, that in equation 1(ii) is in the frequency domain and those in equations 1(iii) and 1(iv) are in the mixed time-frequency domains. It is also seen from equation (1) that the operators H_{tt} , H_{ff} and H_{tf} , H_{ft} are pairs of dual operators respectively.

One method of representing nonlinear systems is by means of functional polynomials^(4,5). This paper develops functional polynomials which represent time-varying nonlinear systems in a dual manner in time, frequency and mixed time-frequency domains as in equation (1). Two sets of functional polynomials are developed so that any element of one set is dual to a certain element of the other set. A functional polynomial is said to be dual to another if both the polynomials have the same number of terms and the corresponding terms of the polynomials form dual representations. The relations between

The research reported herein was supported by the Defence Research Telecommunications Establishment of Canada under the grant no. CD DRB/713003 with Queen's University, at Kingston Ontario and by the National Research Council of Canada under grant no. A4392.

the kernel system functions of the corresponding terms of the various member polynomials of each individual set are also determined. It is seen that some of the kernel system functions are related by single or multi-dimensional Fourier transforms while others are related in more complicated manner. Two other sets of functional polynomials are defined such that any kernel system function of any member polynomial of any individual set is related to the corresponding kernel system function of any other member polynomial belonging to the same set by either one or multi-dimensional Fourier transforms. The kernel system functions are functions of more than one variables. The variables can be grouped into two distinct groups. One group consists of only one variable, either time or frequency and the other group consists of several variables of the same type. In order to make the Fourier transform relations unique a convention is followed that a negative exponential is used when transforming from a time to a frequency variable and a positive exponential is used when transforming from a frequency to a time variable.

Another pair of sets of functional polynomials are defined so that any member polynomial of any individual set is dual to certain other member polynomial of the same set.

These three pairs of sets of functional polynomials are not mutually exclusive. Some of these functional polynomials have been used before and many of them are introduced here for the first time.

REFERENCES

1. P.A. Bello: Time-Frequency Duality, ADCOM Research Report No. 1, August 15, 1962.
2. P.A. Bello: Characterization of Randomly Time-Variant Linear Channels, IEEE Transactions on Communications Systems, December, 1963.
3. A. Gersho: Properties of Time-Varying Linear Systems, M.Sc. Thesis, Cornell University, Ithaca, N.Y., 1961.
4. N. Wiener: Nonlinear Problems in Random Theory, The M.I.T. Press, 1958.
5. A.G. Bose: A Theory of Nonlinear Systems, M.I.T. Technical Report No. 309, 1956.

ESTIMATION OF THE SETTLING TIME OF A CLASS OF TIME-VARYING FEEDBACK CONTROL SYSTEMS

N. AHMED, University of Ottawa, Ottawa, Ontario.

V. GOURISHANKAR, University of Alberta, Edmonton, Alberta.

ABSTRACT

In this paper, a method of estimating the settling time of a class of time-varying feed-back control systems is presented. This class of control systems is described by the integral equation

$$x(t) = \tilde{z}(t) + \lambda \int_{-\infty}^t k(t, \tau) x(\tau) d\tau$$

x and $\tilde{z}$ are n -vector functions and k is a $n \times n$ matrix kernel,

The settling time T^* is defined as the minimum time required by the state of the system to return to within a specified neighbourhood of the initial zero state when it is perturbed from it by a signal either at the input or output terminals. The perturbation is applied at $t=0$ and withdrawn at $t = t_0 > 0$.

A precise mathematical formulation of the problem and an approximate solution using functional analysis techniques are presented. The settling time T^* is shown to be one of the solutions of a simple scalar equation of the form

$$t = a_1 \int_{t_0}^t h(\tau) d\tau + a_2$$

A computational algorithm to determine this settling time is presented. An example illustrates the use of the algorithm.

1. INTRODUCTION

In the case of a feedback control system, particularly a regulator, it is often important that the system, when perturbed from an initial equilibrium state returns to it or to within a specified neighbourhood of it in the shortest possible time. In practice, the exact determination of the "settling time" is very difficult especially if the system is time-varying and/or nonlinear. In such a case it may be worthwhile to obtain an estimate of the settling time. Apart from its use in analyzing a specific system, a knowledge of the approximate settling times of a number of systems will enable the designer to choose the "best" system. As is well known, the designer is frequently required to select the "best" system from among a list of suitable candidates.

To the best knowledge of the authors, no published results on this problem seem to be available. One investigation, which appears to be somewhat related is the work of Letov [1]* concerning the problem of "control quality". Letov's discussion is mostly qualitative and is restricted to time-invariant systems. He does not give any computational procedure.

In this paper the estimation of the settling time of a class of linear time-varying feedback control systems is considered. A precise mathematical formulation for a single input-single output linear system is discussed first. The generalization of the formulation to multi-variable systems is straight forward. An approximate solution using functional analysis as well as a simple computational algorithm are presented. The application of the algorithm is illustrated by means of an example. The extension of the method to nonlinear systems is indicated.

* Numbers in rectangular brackets refer to articles or books listed under references.

2. STATEMENT OF THE PROBLEM

Consider the feedback control system shown in figure 1.

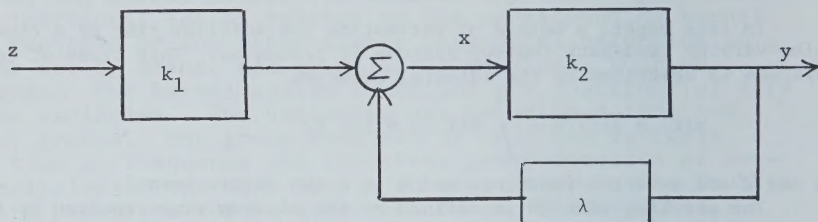


Figure 1

z is the reference input. y is the output. x is the "error". K_1 and K_2 represent linear time-varying systems. They can be considered to be the kernels of linear integral operators. λ is a real number and represents the gain of the feedback loop.

The relationship between the input and the error of the system is described, in general, by the following integral equation

$$x(t) = \int_{-\infty}^t k_1(t, \tau) z(\tau) d\tau + \lambda \int_{-\infty}^t k_2(t, \tau) x(\tau) d\tau \quad (2.1)$$

Equation 2.1 is a linear Volterra integral equation of the second kind. The settling time problem can be stated as follows:

The system shown in figure 1 is perturbed from the initial equilibrium state (say $x=0$) at $t=0$ by a perturbation z and the perturbation is withdrawn at a subsequent time $t=t_0>0$. Given an arbitrary $\epsilon>0$ and that $\max_{t \in [0, t_0]} |z(t)| < \delta_1$ and also that $z(t) \equiv 0$ for t outside the interval $[0, t_0]$, what is the minimum time required for the absolute value of the error signal to reach the ϵ -neighbourhood of the equilibrium state and stay inside the neighbourhood thereafter? If we define a set A as

$$A(t_0, \epsilon) = \{r \in [t_0, \infty] : |x(t)| \leq \epsilon \text{ for all } t \geq r\} \quad (2.2)$$

then the above problem reduces to the determination of $\inf A(t_0, \epsilon)$ if it exists.

Three situations may arise:

- i) the system is unstable, in which case the infimum does not exist
- ii) the system is merely stable, in which case the infimum may exist only for a restricted set of values of $\epsilon>0$. and
- iii) the system is asymptotically stable, in which case the infimum always exists for an arbitrary $\epsilon>0$. We are naturally interested in the last case only.

3. DETERMINATION OF THE SETTLING TIME

A procedure for determining the settling time or in other words, $\inf A(t_0, \epsilon)$ is discussed next.

Taking the absolute value on both sides of equation 2.1, we obtain

$$|x(t)| \leq \int_0^t |k_1(t, \tau)| |z(\tau)| d\tau + |\lambda| \int_0^t |k_2(t, \tau)| |x(\tau)| d\tau \quad t \geq 0 \quad (3.1)$$

Let $z(t) \equiv 0$ for all $t > t_0$ and

$$\text{let} \quad \max_{t \in [0, t_0]} |z(t)| = \delta_1$$

$$\text{and} \quad \max_{t \in [0, t_0]} |x(t)| = \delta_2 \quad (\delta_1)$$

For $t > t_0$, we have

$$|x(t)| \leq \delta_1 \int_0^{t_0} |k_1(t, \tau)| d\tau + |\lambda| \delta_2 \int_0^{t_0} |k_2(t, \tau)| d\tau + |\lambda| \int_{t_0}^t |k_2(t, \tau)| |x(\tau)| d\tau \quad (3.2)$$

We assume that there exist an $\alpha_1 > 0$ and two bounded, continuous non-negative functions $h_1(t)$ and $h_2(t)$ such that

$$|k_1(t, \tau)| \leq e^{-\alpha_1(t-\tau)} h_1(\tau) \quad (3.3)$$

$$\text{and} \quad |k_2(t, \tau)| \leq e^{-\alpha_1(t-\tau)} h_2(\tau)$$

for all τ and t satisfying $0 \leq \tau \leq t \leq \infty$

Then, for all $t > t_0$, inequality 3.2 reduces to

$$\phi(t) \leq \beta_1 + \beta_2 \int_{t_0}^t h_2(\tau) \phi(\tau) d\tau \quad (3.4)$$

where $\phi(t) = |x(t)| e^{\alpha_1 t}$

$$\beta_1 = \int_{t_0}^t e^{\alpha_1 \tau} [\delta_1 h_1(\tau) + |\lambda| \delta_2 h_2(\tau)] d\tau$$

and $\beta_2 = |\lambda|$

Applying Gronwall's lemma [2] to inequality 3.4, we obtain

$$|x(t)| \leq \beta_1 \exp \left\{ \beta_2 \int_{t_0}^t h_2(\tau) d\tau - \alpha_1 t \right\} \text{ for all } t \geq t_0 \quad (3.5)$$

It is clear from 3.5 that $\lim_{t \rightarrow \infty} |x(t)| \rightarrow 0$,

if $\lim_{t \rightarrow \infty} h_2(t) \rightarrow 0$, where $0 \leq C < \alpha_1/\beta_2$. If $h_2(t)$ satisfies this condition then the set $A(t_0, \epsilon)$ defined in 2.2 is non-empty and its infimum is finite. In other words, there exists a finite time T^* , such that $|x(t)| \leq \epsilon$ for all $t \geq T^*$. A constructive existence proof of the foregoing statement is given in the appendix.

Although the existence of $\inf A(t_0, \epsilon)$ has been proved, its determination is not easy. Let us therefore define another set $B(t_0, \epsilon)$ as

$$B(t_0, \epsilon) = \left\{ r \in [t_0, \infty] : x_1(t) \leq \epsilon \text{ for all } t \geq r \right\} \quad (3.6)$$

where

$$x_1(t) = \beta_1 \exp \left\{ \beta_2 \int_{t_0}^t h_2(\tau) d\tau - \alpha_1 t \right\} \quad (3.7)$$

It is clear that $B(t_0, \epsilon) \subset A(t_0, \epsilon)$ and $\inf B(t_0, \epsilon) \geq \inf A(t_0, \epsilon)$. $\inf B(t_0, \epsilon)$ can be easily determined and in most cases it can be considered to be a fairly good estimate of the settling time of the system.

The set $B(t_0, \epsilon)$ can be expressed in another form:

$$B(t_0, \epsilon) = \left\{ r \in [t_0, \infty] : t \geq a_1 \int_{t_0}^t h_2(\tau) d\tau + a_2 \text{ for all } t \geq r \right\} \quad (3.8)$$

where $a_1 = \beta_2/\alpha_1$ and $a_2 = \frac{1}{\alpha_1} \ln(\beta_1/\epsilon)$.

Determination of $\inf B(t_0, \epsilon)$. Case 1. If the inequality

$$t \geq a_1 \int_{t_0}^t h_2(\tau) d\tau + a_2 \quad (3.9)$$

is satisfied for all $t \geq t_0$, then $\inf B(t_0, \epsilon) = t_0$

Case 2. If inequality 3.9 is not satisfied, then

$$\inf B(t_0, \epsilon) = \max_{t \in [t_0, \infty]} (S_1 \setminus S_2) \quad (3.10)$$

where

$$S_1 = \left\{ t \in [t_0, \infty] : t = a_1 \int_{t_0}^t h_2(\tau) d\tau + a_2 \right\} \quad (3.11)$$

and $S_2 \subset S_1$ is defined as

$$S_2 = \left\{ t \in S_1 : t - \eta \geq a_1 \int_{t_0}^{t-\eta} h_2(\tau) d\tau + a_2 \right. \\ \left. \text{for sufficiently small } \eta > 0 \right\} \quad (3.12)$$

Under case 2 we can consider two possibilities. Either set S_2 is empty or not empty. If S_2 is empty then the infimum of B is given by the largest value of t which satisfies the equation

$$t = a_1 \int_{t_0}^t h_2(\tau) d\tau + a_2 \quad (3.13)$$

However, if S_2 is nonempty⁺, then a small quantity $\pm\eta$ can be added to a solution T^* of the equation 3.13 to determine whether or not T^* is an element of S_2 . If T^* is not an element of S_2 , then it may be the desired solution; if T^* is an element of S_2 , then a sufficiently large positive quantity η can be subtracted from T^* to continue the iteration process which is required to solve equation 3.13. This iterative or successive approximation process is described in the course of the proof of a proposition given in the appendix. It is important to observe that whether S_2 is empty or nonempty, an estimate of the settling time of the system is given by one of the appropriate solutions of equation 3.13.

4. AN EXAMPLE

As an illustration of the above technique, let us consider the example of a separable time-varying system (figure 2). $r(t)$ is a known multiplicative disturbance at the input terminals of a time-invariant second order differential system. Comparing figures 1 and 2 we have the following relations.

$$k_1(t, \tau) = \delta(t - \tau) \quad (4.1)$$

$$k_2(t, \tau) = \frac{e^{-\alpha_1(t, \tau)} \sin \alpha_2(t - \tau)}{\alpha_2} r(t) \quad (4.2)$$

where $\alpha_1 = \frac{a}{2}$

$$\alpha_2 = \sqrt{b - \frac{a^2}{4}}, \text{ assuming } b - \frac{a^2}{4} > 0 \quad (4.3)$$

$$\text{Let } r(t) = \left(\frac{\mu_0 + \mu_1 t^2}{1 + t^2} + \mu_2 \cos \omega t \right) \quad (4.4)$$

⁺The set S_2 is nonempty whenever the equation

$$t - a_1 \int_{t_0}^t h_2(\tau) d\tau - a_2 = 0$$

has a continuum of roots and/or has roots of even multiplicity. For example, if both t_i and t_f are in S_1 and $t_i < t_f$ then a necessary condition that the interval $[t_i, t_f] \subset S_2$ is that $h_2(t) \stackrel{i}{=} \frac{1}{a_1}$ for all $t \in [t_i, t_f]$. However, since $h_2(t)$ is related to the kernel of a physical system such a situation will not be very common.

The numerical computations were carried out on an IBM 360/67 computer using Fortran IV language. The following numerical values were used:

$$a = 2; b = 8; \mu = 1, \mu_2 = 1.5; \lambda = -1$$

Different values of μ_0 and ω were used. The value $\lambda = +1$ was also tried to see whether the nature of the feedback had any effect on the settling time. In order to assess the accuracy of the estimate the exact settling time was determined by solving for $x(t)$.

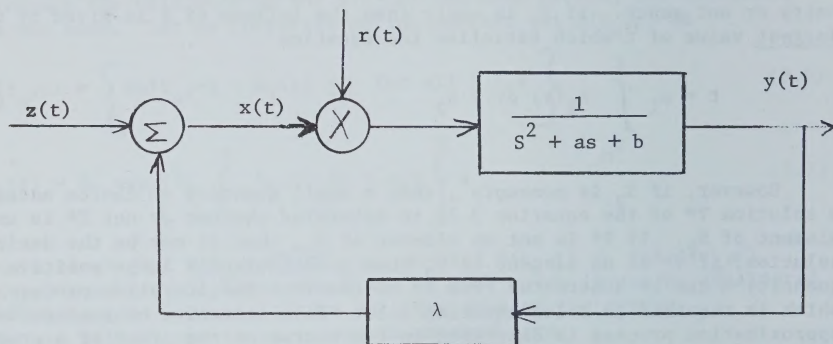


Figure 2

Some of the results are summarized in table 1.

λ	μ_0	ω r.p.s.	Estimated value T^* sec	Exact value T_E sec
-1	1	0.61	6.5	4.5
-1	10	0.61	13.9	4.5
-1	10	3.66	13.6	4.4
+1	1	3.66	6.6	3.7

Table 1

The computational algorithm for T^* seems to yield accurate results for a small value of μ_0 . For large values of μ_0 , the estimate is somewhat pessimistic. However as mentioned at the beginning of the paper, the exact determination of the settling time is often very difficult and tedious and in such cases even a pessimistic estimate is of great value to the designer especially in comparing different systems.

5. NONLINEAR SYSTEMS

The method presented in this paper can be applied to certain nonlinear systems as well. For instance, we can include a zero-memory nonlinear element in the forward loop to precede the linear system represented by K_2 in figure 1. In other words the settling time can be estimated for systems represented by the integral equation

$$x(t) = z(t) + \lambda \int_{-\infty}^t k_2(t, \tau) g(\tau, x(\tau)) d\tau \quad (5.1)$$

for all g satisfying the conditions

$$g(t, 0) = 0 \text{ for all } t \quad (5.2)$$

$$\text{and } |g(t, u)| \leq a(t)|u| \text{ for all } u \in R \quad (5.3)$$

The function $a(t) \geq 0$ is required to be continuous and bounded for all $t \in R$ and $K_2(t, \tau)$ must satisfy a property similar to 3.3 satisfied by $K_2(t, \tau)$ alone in the linear case.

APPENDIX I

Proposition: Let

$$t = a_1 \int_{t_0}^t h_2(\tau) d\tau + a_2 \quad (A.1)$$

For every pair of finite real numbers $a_1 > 0$ and $a_2 \geq 0$, and for every continuous nonnegative function $h_2(\tau)$, if $\lim_{t \rightarrow \infty} h_2(t) \rightarrow c$, for $0 \leq c < 1/a_1$, then $\inf B(t, \epsilon)$ defined by 3.6 exists and is finite. This infimum can be determined from the solution of equation A.1 which is the same as equation 3.13 by a method of successive approximation.

Proof: Let us define a sequence $\{T_n\}$ by

$$T_{n+1} = a_1 \int_{t_0}^{T_n} h_2(\tau) d\tau + a_2 \quad (A.2)$$

($n = 0, 1, 2, \dots$)

For an arbitrary choice of T_0 , the sequence T_n may converge to any of the intermediate solutions of equation A.1. To determine $\inf B$, we may choose T_0 as follows: since $\lim_{t \rightarrow \infty} h_2(t) \rightarrow c < 1/a_1$, there exists for every $c' \in (c, 1/a_1)$, a $T_0(c') < \infty$ such that $h(t) < c'$ for all $t \geq T_0(c')$. Let us choose a value for T_0 such that $T_0(c') \leq T_0 < \infty$. It is clear that $h(t) < c'$ for all $t \geq \min(T_0, \inf B)$. Therefore

$$|T_{n+1} - T_n| \leq \alpha |T_n - T_{n-1}| \text{ for } n \geq 0$$

where $\alpha = a_1 c' < 1$. For every integer $p \geq 1$, we have

$$\begin{aligned} |T_{n+p} - T_n| &\leq \sum_{k=n}^{n+p-1} |T_{k+1} - T_k| \\ &\leq |T_1 - T_0| \sum_{k=n}^{n+p-1} \alpha^k \\ &\leq \frac{|T_1 - T_0|}{1 - \alpha} \alpha^n \text{ for all } n \geq 0. \end{aligned}$$

Thus $\lim_{n \rightarrow \infty} |T_{n+p} - T_n| \rightarrow 0$ for every $p \geq 1$ and therefore $\{T_n\}$ is a Cauchy sequence in $[t_0, \infty)$. Hence T_n converges to a limit point $T^* \in [t_0, \infty)$. That T^* is a solution of equation A.1 is proved as follows:

Let us define $g(T)$ by

$$g(T) = a_1 \int_{t_0}^T h_2(\tau) d\tau + a_2 \quad (A.3)$$

Then

$$\begin{aligned} |T^* - g(T^*)| &= |T^* - g(T_n) + g(T_n) - g(T^*)| \\ &\leq |T^* - T_{n+1}| + a_1 \left| \int_{T^*}^{T_n} h_2(\tau) d\tau \right| \end{aligned}$$

Since the left hand side is independent of n the inequality is true for any n . Since $T_n \rightarrow T^*$ and $h_2(t)$ is continuous and bounded we have $T^* = g(T^*)$. If $T^* \in S_2$ then the process of iteration is continued. Let us therefore consider the other possibility, namely $T^* \in S_1 \setminus S_2$. (S_1 and S_2 are defined earlier in section 3).

Since every Cauchy Sequence is bounded, T^* is bounded and also

$$T^* \leq \frac{a_2' - \alpha T_0}{1 - \alpha} < \infty$$

where
$$a_2' = a_2 + a_1 \int_{t_0}^{T_0} h_2(\tau) d\tau < \infty$$

That T^* is actually the desired $\inf B$ is proved as follows: Suppose T^* is not the $\inf B$, then there exists a $t^* > T^*$ such that

$$t^* < a_1 \int_{t_0}^{t^*} h_2(\tau) d\tau + a_2$$

If $\{T_n\}$ is a nondecreasing sequence then $h_2(t) < c'$ for all $t > T^*$. Hence $t^* \leq T^* + \alpha(t^* - T^*)$ implying that $t^* < T^*$. This leads to a contradiction. If $\{T_n\}$ is a nonincreasing sequence then $t^* > T^*$ implies also that $t^* > T_0$ because

$$t > a_1 \int_{t_0}^t h_2(\tau) d\tau + a_2 \quad \text{for all } t \in [T^*, T_0].$$

Hence from the inequality -

$$t^* < a_1 \int_{t_0}^{t^*} h_2(\tau) d\tau + a_2$$

it follows that

$$t^* < T_1 + \alpha(t^* - T_0) \leq T_0 + \alpha(t^* - T_0)$$

which implies that $t^* < T_0$, thus leading to a contradiction. Therefore T^* is actually the desired $\inf B$.

APPENDIX II

Multivariable systems: The technique described in this paper applies to multivariable systems also. If $z(t)$ is a m -vector and $x(t)$ is an n -vector, the kernels k_1 and k_2 are $n \times m$ and $n \times n$ matrices respectively. The elements of k_1 and k_2 are functions of t and τ over the triangle $\Delta = \{t, \tau \mid -\infty < \tau \leq t < \infty\}$. λ is a scalar or a diagonal matrix with elements $(\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n)$. The symbol $|\cdot|$ may be interpreted as any combination of the standard norms defined in Euclidean n -space. For example $|x(t)|$ may be taken as either

$$(i) \quad |x(t)| = \max_i |x_i(t)| \quad \text{or} \quad (ii) \quad |x(t)| = \sqrt{\sum_{i=1}^n |x_i(t)|^2}$$

If the elements of $k(k_1 \text{ or } k_2)$ are represented by $k_{ij}(t, \tau)$, then $|k(t, \tau)|$ may be chosen to be either

$$(i)' \quad |k(t, \tau)| = \max_{ij} |k_{ij}(t, \tau)|$$

$$\text{or} \quad (ii)' \quad |k(t, \tau)| = \sqrt{\sum_{ij} |k_{ij}(t, \tau)|^2}$$

for all $t, \tau \in \Delta$.

The first set of norms (i) and (i)' may yield a sharper estimate of the settling time T^* .

ACKNOWLEDGEMENT

The authors would like to express their appreciation to Mr. A. Salama, for his valuable assistance in the computation of the numerical results.

REFERENCES

1. Letov, A.M., Stability in Nonlinear Control Systems. Princeton University Press 1961. (pp. 233 - 252).
2. Zadeh, L.A. and Desoer, C.A., Linear System Theory. McGraw-Hill Book Company 1963. (p. 374).

EXACT AND APPROXIMATE SOLUTIONS FOR LINEAR TIME-VARYING SYSTEMS

HOOSHANG HEMAMI

Department of Electrical Engineering
The Ohio State University
Columbus, Ohio

ABSTRACT

In general closed form solutions for linear time-varying deterministic systems do not exist. Consequently one starts with an initial solution and adds corrective terms by iteration or perturbation; and in most cases of practical interest the rate of convergence of this approximation depends on the choice of the initial solution.

In this paper, criteria are discussed that permit better selection of an initial solution. These criteria depend on, 1) how close the system is to classes of linear time-varying systems for which closed form solutions exist, and 2) how closely the state transition matrix of the system can be approximated by response of the time-varying system to special inputs for which closed form responses exist.

First, some known classes of time-varying systems for which closed form solutions exist are discussed, and some tests that can be performed to verify whether the system belongs to any of these classes are outlined. Then a natural basis is introduced in the state space that renders closed form solutions for classes of inputs. As a consequence of this basis, concept of controllability of time-varying systems is introduced in a natural way.

Finally, steps to be taken, and the amount of prior computations for the choice of the initial solutions are outlined.

1. Introduction

The problem is to solve the set of first order equations

$$\begin{aligned}\dot{X} &= A(t) X(t) + B(t) U(t) & X(0) &= X_0 \\ Y(t) &= C^T(t) X(t)\end{aligned}\tag{1}$$

where X , U , and Y are respectively a n vector and two scalars. $A(t)$, $B(t)$, and $C(t)$ are compatible matrices with continuous and continuously differentiable elements for all $t > 0$. For our purpose, it is sufficient to derive the solution for the homogeneous system, namely,

$$\dot{X} = A(t) X(t) \qquad X(t) = X_0 \tag{2}$$

In general, closed form solutions do not exist for Eq. (2) except for special classes of operators $A(t)$. The operators considered here will be characterized by eigenvalues and eigenvectors if the eigenvalues are distinct, or by invariant subspaces and the eigenvalues associated with these invariant subspaces if the eigenvalues are repeated. Such characterization provides a natural decomposition of the operator and consequently renders geometrical insight into the structure of the operator.

An invariant subspace $S(t)$ of an operator $A(t)$ is such that if a vector $Z(t) \in S(t)$ for all t then $A(t) Z(t) \in S(t)$ for all t .

The general operator $A(t)$ with distinct eigenvalues has time-varying eigenvalues and time-varying eigenvectors. As two special classes, one can consider: 1) operators that have constant eigenvectors and only their eigenvalues are functions of time, and 2) operators that have constant eigenvalues and time-varying eigenvectors. The first class and a subclass of the second class render closed form solutions for Eq. (2). If the operator has repeated eigenvalues, the above two special classes become: 1) operators that have constant invariant subspaces and only the eigenvalues are functions to time, and 2) operators that have constant eigenvalues, and time-varying subspaces of a particular type to be discussed below. For both of these multiple eigenvalue cases closed form solutions exist.

For simplicity we assume all eigenvalues and eigenvectors are real.

2. Constant Invariant Subspaces

The class of time-varying systems that have constant invariant subspaces are well known.^{1,2,3} For this class only the eigenvalues are functions of time. The solution of Eq. (2) for this class is exponential, namely, the state transition matrix of Eq. (2) is

$$\phi(t, 0) = e^{\int_0^t A(\tau) d\tau} \quad (3)$$

In order for $A(t)$ to have constant invariant subspaces it is necessary and sufficient that

$$A(t_1) A(t_2) - A(t_2) A(t_1) = 0, \quad t_1, t_2 \geq 0 \quad (4)$$

It can also be shown that if Eq. (4) is satisfied

$$A(t) \int_0^t A(\tau) d\tau = \int_0^t A(\tau) d\tau A(t) \quad (5)$$

while the inverse is not necessarily true.³ Conditions (3) and (4) are satisfied whether $A(t)$ has n distinct eigenvalues with n one-dimensional invariant subspaces (eigenvectors) or $A(t)$ has any multiplicity of eigenvalues, and consequently combinations of invariant subspaces of any dimensionality.

In addition, for this class, the concepts of observability and controllability can be studied as in the time-invariant systems since the subspaces are invariant with time.²

3. Constant Eigenvalues

In the second class, consider an operator $A(t)$ whose eigenvalues are constants, and whose eigenvectors are functions of time. Without loss of generality, we assume $A(t)$ to be of the class

$$A(t) = M(t) \Gamma M^{-1}(t) \quad (6)$$

where $M(t)$ is $n \times n$, and nonsingular and Γ may be diagonal with distinct elements or any of the conceivable Jordan forms (repeated diagonal elements and ones above the diagonal line).

One simple set of necessary and sufficient conditions for an operator to have constant eigenvalues is

$$\text{Trace } A^i(t) = \sum_{j=1}^n \gamma_j^i = \text{constant} \quad i = 1, 2, \dots, n \quad (7)$$

Consider a subclass of operators (6) for which $M(t) = e^{Pt}$ where P is a constant matrix:

$$A(t) = e^{Pt} \Gamma e^{-Pt} \quad (8)$$

A necessary and sufficient condition for $A(t)$ to satisfy Eq. (8) is that $A(t)$ satisfies the following differential equation

$$\dot{A}(t) = PA(t) - A(t)P \quad (9)$$

If $A(t)$ satisfies Eq. (9), it can be verified that the state transition matrix associated with Eq. (2) is

$$\phi(t, 0) = e^{Pt} e^{(\Gamma - P)t} \quad (10)$$

In order to test whether a given $A(t)$ satisfies Eq. (9) or not, one can solve the latter as a matrix equation for P .

The simplest test appears to be the following. Assume $A(t)$ is an analytic function of time and select a convenient time instant t_1 .

Solve equation

$$P_1 A(t_1) - A(t_1) P_1 = \dot{A}(t_1) \quad (11)$$

for P_1 . The solution is involved and is discussed in part 7. Once P_1 is derived, one can verify whether Eq. (9) is satisfied or not for P_1 .

4. Kronecker Sum

Pease⁵ has shown that if the systems

$$\dot{X}(t) = A_1(t) X(t) \quad \dim = m \quad (12)$$

$$\dot{Y}(t) = E_1(t) Y(t) \quad \dim = n \quad (13)$$

have closed form solutions, then the system

$$\dot{Z}(t) = C_1(t) Z(t) \quad (14)$$

has a closed form solution if the Kronecker sum of A_1 and E_1 is equal to C_1 :

$$C_1(t) = A_1(t) \times I_n + I_m \times E_1(t) \quad (15)$$

where $\times$ means the direct product, and in which case the state transition matrix of system (14) is the direct product of the state transition matrices of systems (12) and (13). This class is the same as in part 2 since $C_1(t)$ commutes with itself.

5. Riccati Condition

If operator A of Eq. (2) satisfies the Riccati Equation:

$$\dot{A} + A^2 = PA \quad (16)$$

where $P(t)$ is an element of one of the two classes discussed before, it can be shown that $\dot{X}(t)$ has a closed form solution, and consequently $X(t)$ can be derived in closed form.⁶ As an example, assume P is constant, then

$$X(t) = [I + P^{-1}(e^{Pt} - I) A(0)] X(0) \quad (17)$$

6. Approximate System Response

6.1. Particular Solution. Consider again system of Eq. (1), and let $X(0) = 0$. We can think of the state response of the system, namely, $X(t)$ as a sum of n independent modes

$$X(t) = \sum_{i=0}^{n-1} \alpha_i(t) B_i(t) \quad (18)$$

where $B_i(t)$ are n known independent vectors for all time $0 < t < \infty$ and $\alpha_i(t)$ is a differentiable scalar function of time. This is a natural extension of the approach in reference 2 for constant systems. If $B_i(t)$'s are n independent vectors and $\alpha_i(t)$ is not identically zero, then the system can be controlled--that is--it has controllable response along the n independent vectors and consequently in the whole space. Substituting for $\dot{X}$ and X in Eq. (1) from Eq. (18), one obtains:

$$\sum_{i=0}^{n-1} \dot{\alpha}_i(t) B_i(t) + \sum_{i=0}^{n-1} \alpha_i(t) \dot{B}_i(t) = \sum_{i=0}^{n-1} \alpha_i(t) A(t) B_i(t) + B(t) u(t) \quad (19)$$

Now coefficients of independent vectors on both sides of Eq. (3) can be set equal to each other. However, there may arise two difficulties because B_i 's are functions of time:

- 1) $\dot{B}_i(t)$ may not be related to B_i 's in a known manner.
- 2) $A(t) B_i(t)$ may not be related to B_i 's in a known manner.

Since B_i 's are at our disposal, they can be defined as

$$\begin{aligned} B_0 &= B \\ B_1 &= AB_0 - \dot{B}_0 \\ &\dots \dots \dots \\ B_{n-1} &= AB_{n-2} - \dot{B}_{n-2} \end{aligned} \quad (20)$$

Assuming that they are independent, and equating coefficients of B_i on both sides of Eq. (19) we derive the following relations among α_i 's:

$$\begin{aligned}
\dot{\alpha}_0(t) &= u(t) \\
\dot{\alpha}_1(t) &= \alpha_0(t) \\
&\dots \dots \dots \\
\dot{\alpha}_{n-1}(t) &= \alpha_{n-2}(t)
\end{aligned}
\tag{21}$$

But the following term on the right hand side of Eq. (18) will not be accounted for.

$$R(t) = \alpha_{n-1}(t) [AB_{n-1} - \dot{B}_{n-1}] = \alpha_{n-1}(t) B_n(t) \tag{22}$$

where $B_n(t)$ cannot be independent from $B_0, \dots, B_{n-1}$.

It has been proven by Silverman⁷ that for controllable systems $B_0(t), \dots, B_{n-1}(t)$ are independent. Assume for the time being that

$$R(t) \equiv 0 \tag{23}$$

Then Eqs. (21) can be integrated to give

$$\begin{aligned}
\alpha_0(t) &= \int_0^t u(t_1) dt_1 \\
\alpha_1(t) &= \int_0^t dt_1 \int_0^{t_1} u(t_2) dt_2 \\
&\dots \dots \dots \\
\alpha_{n-1}(t) &= \int_0^t dt_1 \int_0^{t_1} dt_2 \int_0^{t_2} \dots \int_0^{t_{n-1}} u(t_n) dt_n
\end{aligned}
\tag{24}$$

Thus Eqs. (20) and (24) can be substituted in Eq. (18) for the exact solution.

6.2 Homogeneous Solution. This same method can be used for the response of System (2) and the state transition matrix derived in closed form.

Let the response due to an initial state X_0 be

$$X(t) = \sum_{i=0}^{n-1} \beta_i(t) B_i(t) \tag{25}$$

Note that now the choice of $B_0(t)$ is arbitrary. Substitution of Eq. (25) in

$$\dot{X} = A(t) X(t) \quad X(0) = X_0 \tag{26}$$

renders

$$\sum_{i=0}^{n-1} \dot{\beta}_i(t) B_i(t) + \sum_{i=0}^{n-1} \beta_i(t) \dot{B}_i(t) = \sum_{i=0}^{n-1} \beta_i(t) B_i(t) \tag{27}$$

it follows that

$$\begin{aligned}\dot{\beta}_0(t) &= 0 \\ \dot{\beta}_1(t) &= \beta_0(t) \\ &\dots \dots \dots \\ \dot{\beta}_{n-1}(t) &= \beta_{n-2}(t)\end{aligned}\tag{28}$$

or

$$\begin{aligned}\beta_0(t) &= C_0 \\ \beta_1(t) &= C_0 t + C_1 \\ &\dots \dots \dots\end{aligned}\tag{29}$$

$$\beta_{n-1}(t) = C_0 \frac{t^{n-1}}{(n-1)!} + C_1 \frac{t^{n-2}}{(n-2)!} + \dots + C_{n-2} t + C_{n-1}$$

Recalling that

$$X_0 = \sum_{i=0}^{n-1} \beta_i(0) B_i(0)\tag{30}$$

it follows that

$$X_0 = \sum_{i=0}^{n-1} C_i B_i(0)\tag{31}$$

Let $F_i(0)$ [$i = 0, \dots, n-1$] be a set of constant vectors such that

$$\begin{aligned}F_i^T(0) B_j(0) &= \delta_{ij} = 1 && \text{if } i = j \\ &0 && \text{if } i \neq j\end{aligned}\tag{32}$$

Then

$$C_i = F_i^T X_0\tag{33}$$

$$X(t) = \sum_{i=0}^{n-1} B_i(t) \sum_{j=0}^i C_{i-j} \frac{t^j}{j!}\tag{34}$$

or finally

$$X(t) = \sum_{i=0}^{n-1} B_i(t) \sum_{j=0}^i \frac{t^j}{j!} F_{i-j}^T(0) X_0\tag{35}$$

or

$$\phi(t, 0) = \sum_{i=0}^{n-1} B_i(t) \sum_{j=0}^i \frac{t^j}{j!} F_{i-j}^T(0)\tag{36}$$

$$\phi(t,0) = \begin{bmatrix} . \\ . \\ B_0(t) \dots B_{n-1}(t) \\ . \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t & 1 & 0 \\ \dots\dots\dots \\ \frac{t^{n-1}}{(n-1)!} & \dots 1 \end{bmatrix} \begin{bmatrix} \dots F_0^T(0) \dots \\ . \\ . \\ \dots F_{n-1}^T(0) . \end{bmatrix} \quad (37)$$

If Eq. (23) is not satisfied, neither the particular solution nor the homogeneous solution (36) is exact. They are approximate solutions, and $R(t)$ is an indication of how far these solutions are from the exact solution.

Actually Eq. (36) is an exact solution of the system

$$\dot{X} = A(t) X(t) - \beta_{n-1}(t) B_n(t) \quad X(0) = X_0 \quad (38)$$

This means $\phi(t,0)$ of Eq. (37) is the exact solution of Eq. (38) and not of Eq. (2). The input term $\beta_{n-1}(t) B_n(t)$ is of course proportional to input X_0 .

The latter fact can be used to find exact solutions for driven linear time-varying systems for special classes of inputs: suppose $\phi_1(t,0)$ is the solution of System (38) for $X(0) = X_1$ and a given $R_1(t) = \beta_{n-1}(t) B_n(t)$ resulting from application of steps (25-37). Suppose $\phi_2(t,0)$ is the solution of System (38) for $X(0) = X_2$ and a different $R_2(t) = \beta_{n-1}(t) B_n(t)$ resulting from a different application of steps (25-37).

The total response of the system

$$\dot{X} = A(t) X(t) - (R_1(t) + R_2(t)) \quad X(0) = X_1 + X_2 \quad (39)$$

is given by

$$X(t) = \phi_1(t,0) X_1 + \phi_2(t,0) X_2 \quad (40)$$

The same procedure can be applied to a system drive with $R_1(t) + R_2(t) + \dots + R_m(t)$.

7. Discussion

In Sections 2-5 classes of linear time-varying operators were discussed for which closed form solutions for the state transition matrix exist. This is a modest start in classification of such operators, and by no means exhausts all operators with closed form solution. Two questions of immediate importance can be asked: 1) What geometrical and algebraic properties govern the structure of such operators? and 2) What simple identification schemes can be used to identify whether an operator possesses or does not possess closed form solutions?

As far as the first question is concerned, an adequate explanation exists in terms of constant eigenvectors and constant eigenvalues for the first two classes. Also for the first two classes a reasonably good foundation exists in terms of Lie Algebras and exponential solutions of differential equations.^{8,9,10,11} However, Lie Algebras do not seem to explain the Riccati condition.

The identification of operators with closed form solution depends on

how well the structure is understood. Within the presentation of this paper, condition (4) is fairly simple to verify for the first class. Condition (11) is more difficult to verify. The solution of Eq. (11) is not easy, and involves construction of the n projection operators associated with the one-dimensional constant subspaces of the operator $A(t_1)$. For necessary and sufficient conditions for the existence of a solution B_1 and the derivation of B_1 , see Rosenbloom.¹² If $A(t_1)$ has invariant subspaces of higher dimensions the same procedure of Rosenbloom can be used except that there are fewer than n projection operators and construction of the latter projection operators is more involved. It is to be recalled that the construction of these projectors requires either knowledge of eigenvalues of $A(t_1)$ or knowledge of the eigenvectors and corresponding reciprocal basis vectors of $A(t_1)$.

Verification of Eq. (15) for a given $C_1(t)$ is more involved. No easy methods have been developed to test C_1 . Even if $C_1(t)$ can be decomposed, there may not exist closed form solutions for Systems A_1 and B_1 thus derived. The Riccati condition (16) also is difficult to verify. At least it requires inversion of $A(t)$ or $A(t_1)$ where t_1 is a given time.

Despite these difficulties associated with identification of such operators, a given operation can be checked for closeness to one of such classes on an interval of interest. A faster rate of convergence of a series approximation and the saving of computations and time may justify some initial investigation of the operator.

In Section 6 a method of approximation of state transition matrix was discussed that uses a naturally constructed basis. It was shown that for particular classes of inputs a closed form solution for the system can be derived without knowledge of the exact state transition matrix. In addition this approach gives a justification for the construction of the sequence of vectors that have been lately proposed for controllability analysis.

8. Conclusions

Several classes of time-varying systems for which closed form solutions for the state transition matrix exist were discussed. Much remains to be done to characterize and identify all such operators. The knowledge of the exact solutions can help in the selection of an initial state transition matrix to be perturbed or iterated. The questions of rate of convergence and stability of these approximations require additional research. A certain approximation to the state transition matrix was discussed that uses a natural basis and can render closed form solution for some particular inputs.

ACKNOWLEDGMENT

This research was supported by The Ohio State University College of Engineering.

The author would like to thank Professor Frank M. Brown of the Air Force Institute of Technology for stimulating discussions.

REFERENCES

1. Kinariwala, B. K., "Analysis of Time-Varying Networks," IRE Inter. Conv. Record 1961, Part 4, pp. 268-276.
2. Hemami, H. and Weimer, F. C., "A Geometrical Exposition of State Space Techniques," Proceedings, First Asilomar Conference on Circuits and Systems, 1967, pp. 552-561.
3. Martin, J.F.P., "Some Results on Matrices which Commute with their Derivatives," SIAM J. Applied Math, Vol. 15, No. 5, September 1967, pp. 1171-1183.
4. Pease, III, M. C., Methods of Matrix Algebra, Academic Press, New York, 1965, Chapter 13.
5. Ibid, Chapter 14, Part 3.
6. Hemami, H., "On a Class of Time-Varying Systems," to appear in IEEE Transactions AC, October 1968.
7. Silverman, L. M. and Meadows, H. E., "Controllability and Observability in Time-Variable Systems," SIAM J. Control, Vol. 5, No. 1, February 1967, pp. 64-73.
8. Magnus, W., "On the Exponential Solution of Differential Equations for a Linear Operator," Comm. Pure Appl. Math., 7, 1954, pp. 649-673.
9. Norman, E. and Wei, J., "Lie Algebraic Solution of Linear Differential Equations," J. Mathematical Phys., 4, 1963, pp. 336-363.
10. Wei, J. and Norman, E., "On Global Representations of the Solutions of Linear Differential Equations as a Product of Exponents," Proc. Amer. Math. Soc., 15, 1964, pp. 327-334.
11. Wichmann, E., "Note on the Algebraic Aspect of the Integration of a System of Ordinary Linear Differential Equations," J. Mathematical Phys., 2, 1961, pp. 876-880.
12. Rosenbloom, M., "On the Operator Equation $BX - XA = Q$," Duke Mathematical Journal, Vol. 23, 1956, pp. 263-269, Theorem 4-6.

S.V. SANKARAN and S.A. MUSA
The Moore School of Electrical Engineering
University of Pennsylvania, Philadelphia, Pa.

This paper presents a new approach for the application of transform methods to linear time-varying systems. Narendra [1] has defined a system function $H(\lambda)$ independent of time and has demonstrated its use with integral transforms for time-varying systems. In this paper a more general system function--one which explicitly contains a time variation--namely $H_1(\lambda) H_2(t)$, is proposed. This system function may be defined as follows. If the input to a time-varying linear system is the "elementary" signal $k(t, \lambda)$ the output in the time-domain is $H_1(\lambda) H_2(t) k(t, \lambda)$. This function is shown to exhibit greater capabilities than the ones hitherto employed. Methods of obtaining suitable transforms for various classes of systems by suitable choices of $H_1(\lambda)$ and $H_2(t)$ are discussed in this paper.

For the general nth order system

$$M(p, t) v(t) = x(t) u(t) \quad (1)$$

where $M(p, t) = b_n(t) \frac{d^n}{dt^n} + b_{n-1}(t) \frac{d^{n-1}}{dt^{n-1}} + \dots + b_1(t) \frac{d}{dt} + b_0(t)$, the kernel $k(t, \lambda)$ is given by the equation

$$\left[M(p, t) - \frac{x(t)}{H_1(\lambda) H_2(t)} \right] [H_2(t) k(t, \lambda)] = 0 \quad (2)$$

The choice of $H_1(\lambda)$ and $H_2(t)$ in general depends upon the coefficients of the system equation and is made so that $k(t, \lambda)$ may be simple and resemble a known transform kernel.

For the first order systems

$$\left[\frac{d}{dt} + c(t) \right] v(t) = x(t) u(t) \text{ and } \left[\frac{d}{dt} + c(t) \right] v(t) = x(t) \frac{du}{dt},$$

it is shown that the Laplace transform is applicable after a suitable change of the independent variable.

If the general system equation (1) can be expressed in the form

$$L(p, t) [f(t) v(t)] = x(t) u(t), \text{ where } L(p, t) = a_n(t) \frac{d^n}{dt^n} + a_{n-1}(t) \frac{d^{n-1}}{dt^{n-1}} + \dots + a_0(t),$$

then it is shown that the kernels and transform can be determined for certain forms of $L(p, t)$ --second order as well as higher orders.

For the second order system

$$\left[a(t) \frac{d^2}{dt^2} + b(t) \frac{d}{dt} + c(t) \right] [f(t) v(t)] = x(t) u(t) \quad (3)$$

the following cases are considered:

(i) When $a(t)$, $b(t)$, $c(t)$ are just constants a, b, c the Laplace transform is shown to apply.

(ii) When $a(t) = a_2 t^2$, $b(t) = a_1 t$, $c(t) = a_0$, the Mellin transform applies.

(iii) If $a(t) = 1$, $b(t) = 1/t$, $c(t) = \frac{-v^2}{t^2}$ the Hankel transform is applicable.

(iv) If $a(t) = 1$, $b(t) = \frac{2v+1}{t}$, $c(t) = 0$ the Meijer transform is applicable.

The idea of the second order systems (i) and (ii) leading to Laplace and Mellin transforms is shown also to extend to n th order systems with similar coefficients. By substituting $f(t) = t^\alpha$, $e^{\alpha t}$, ... new systems with power of t , exponentials, etc. in the coefficients may be obtained. Narendra's method is of course applicable to the above system if $f(t) v(t)$ is replaced by $v_1(t)$.

Another interesting way of treating time-varying systems using the new system function is when the operator $M(p, t)$ of the system (1) can be expressed as the product of n first order linear operators of the form

$$\left[a(t) \frac{d}{dt} + b(t) \right].$$

The n th order system may then be represented as a series cascade of n simple first order systems each of which can be treated by simple Laplace transform with suitable change of variable. Some systems of this type are discussed.

In short an attempt is made to prove the greater potential of the proposed system function. It is shown that systems previously treated by other methods can be considered as particular cases of more general systems treated by the new method. Higher order systems can be directly handled even when they are not a result of cascading identical simple lower order systems.

References:

1. K.S. Narendra, "Integral Transforms for a Class of Time-Varying Linear Systems", IRE Trans. on A.C., AC-6, 3, pp. 311-318, September 1961.

TRUNK AND TREE SEARCHING PROPERTIES OF THE FANO SEQUENTIAL DECODING ALGORITHM*

JAMES L. MASSEY AND MICHAEL K. SAIN
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

Fano's sequential decoding algorithm is widely recognized as the most practical procedure to search the value tree determined by a tree code (usually a convolutional code) and a received sequence. A trunk is defined as a tree consisting only of a series connection of branches. The trunk searching properties of the Fano algorithm are simply determined. It is shown that the tree search properties of the algorithm follow immediately from the trunk search properties. This approach leads to simple proofs of well-known properties of the Fano search and also uncovers some new and interesting properties.

I. INTRODUCTION

Sequential decoding, introduced by Wozencraft (1) a decade ago, has become widely recognized as an eminently practical technique for communicating over a variety of communications channels in situations where reliable data is required but a small deletion rate is acceptable. This deletion rate stems from the variability of the decoding time per digit which is also the source of the primary benefits of sequential decoding. Except during infrequent periods of severe and protracted channel disturbances, the decoding time is very small and reliable decisions are made. During the periods of high disturbance, the decoding time increases quite sharply so that decoding cannot be completed thus resulting in "erasures" rather than decoding "errors". The sequential decoding algorithm given by Wozencraft has undergone several refinements, culminating in the ingenious algorithm given by Fano (2). There are good grounds for believing that no significant further improvement is possible. It is our purpose here to present a new way of looking at the Fano algorithm which appears to offer significant conceptual simplification over other approaches that have been given. Our approach permits simple proofs of several known properties of the algorithm and leads naturally to some interesting new properties.

Any sequential decoding algorithm is a set of rules for progressing from the origin to a terminal node of a value tree based on examination only of those nodes connected to the presently occupied node. A value tree, an example of which appears in Fig. 1, is just a rooted tree in which each node is assigned some real number, with the root node, or origin, always assigned the value zero. In communications applications, the node values are the sum of "metric" values assigned to each of the branches leading from the origin to that node, but the details of such assignments do not concern us here. The basic goal of a sequential decoding algorithm is, in the case when one of the terminal nodes has a value greatly exceeding the others, to move from the origin to that terminal node of greatest value in the fewest number of moves.

* This research was supported by the National Aeronautics and Space Administration (NASA Grant NGL 15-004-026) in liaison with the Flight Data Systems Branch of the Goddard Space Flight Center.

II. TRUNK SEARCHING PROPERTIES

The flowchart of the Fano sequential decoding algorithm is given in Fig. 2. "Better" and "worse" refer to greater and less node values respectively, with ties resolved in any consistent manner. The left half of this flowchart is the operative section for forward moves. Whenever the best (or next best when entering via the dotted box) node ahead lies on or above the current threshold T , a forward move is made to that node. If this is a first visit, then the threshold is "tightened" by increasing it in increments of a fixed positive parameter Δ until another increase would cause T to exceed the node value V . Hence, after tightening, the threshold value is $\lceil \frac{V}{\Delta} \rceil \Delta$ where here and hereafter $\lceil x \rceil$ denotes the largest integer not exceeding x . On second and subsequent visits, the threshold is not tightened and an immediate attempt is made to move forward again.

When further forward progress is blocked, the right half of the flowchart becomes the operative section for determining what backward moves are required. Understanding of this section is hindered by the provision, indicated by the dotted blocks in Fig. 2, to move forward on alternate branches when a backward move is made to some node from other than the worst node stemming forward from the former. To avoid this difficulty, we now restrict our attention to value trunks, as in Fig. 3, rather than more general value trees. A value trunk is simply a linear value tree. In a value trunk, a backward move is always from the "worst" node so that the dotted boxes in Fig. 2 may be entirely ignored. In this case, we see from Fig. 2 that--when forward progress is blocked--a retreat is made through all previous nodes on or above the current threshold and the threshold then decremented by Δ . Moreover, we see further that forward moves through all these same nodes immediately follow without tightening of the threshold since the nodes have been visited before. The net effect then is that whenever a backward look is made from some node with threshold T_0 , the next forward look is made from that node with threshold $T_0 - \Delta$. But this would be exactly the same effect as occurs simply when the previous node is below T_0 . Hence, we have proved:

Property 1: With respect to all looks and moves rightward of node V_i , the trunk of Fig. 4 is equivalent to that of Fig. 3.

We remark that "equivalence" here and hereafter means the same set of looks and moves, in the same sequence, and with the same thresholds. The starting threshold in Fig. 4 follows from the fact that the first forward look from V_i occurs immediately after the first visit to V_i and we have already seen that the threshold is set to $\lceil \frac{V_i}{\Delta} \rceil \Delta$ when the first visit is made.

The next property is equally fundamental and only slightly less obvious.

Property 2: With respect to all looks and moves leftward of node V_j , the trunk of Fig. 5 is equivalent to that of Fig. 3.

Proof: Consider first the special case where $j = L-2$ so that we claim only that nodes V_{L-1} and V_L can be replaced by a node of their minimum value m . It may be readily verified from Fig. 2 that if a forward look with $T \leq m$ is made from V_{L-2} , then the algorithm will terminate for both trunks without passage back through V_{L-2} . Conversely, with $T > m$, for both trunks the next action at V_{L-2} will be a backward look at the same threshold T . Hence for $j = L-2$, the trunks are equivalent as claimed. For $j < L-2$, we argue as follows. V_{L-1} and V_L may be combined as a single node of value

$\min(V_{L-1}, V_L)$ as already shown. Then this node and V_{L-2} may be combined as a node of value $\min(V_{L-2}, \min(V_{L-1}, V_L)) = \min(V_{L-2}, V_{L-1}, V_L)$. Iteration of this argument proves the property as claimed.

Combining properties 1 and 2, we obtain:

Theorem 1: For all looks and moves between nodes V_i and V_{i+1} , the trunk of Fig. 6 is equivalent to that of Fig. 3.

Application of the flowchart of Fig. 2 to the simple trunk of Fig. 6 makes it possible to deduce all the following trunk search Properties 3-6 by inspection:

Property 3: The first forward look from V_i to V_{i+1} is made with $T = \lceil \frac{V_i}{\Delta} \rceil \Delta$. Each subsequent forward look is made with T less by Δ than the previous. The final forward look is made with $T = \lceil \frac{\min(V_i, V_{i+1}, \dots, V_L)}{\Delta} \rceil \Delta$.

Property 4: The first forward move from V_i to V_{i+1} is made with $T = \lceil \frac{\min(V_i, V_{i+1})}{\Delta} \rceil \Delta$.

The next property which is an immediate consequence of properties 3 and 4 is significant in that a "forward look" is perhaps the best measure of a single "computation" of the algorithm. Of course, the number of backward moves made from V_i is one less than the number of forward moves made from V_i .

Property 5: The number of forward looks made from V_i to V_{i+1} is given by $\lceil \frac{V_i}{\Delta} \rceil - \lceil \frac{\min(V_i, V_{i+1}, \dots, V_L)}{\Delta} \rceil + 1$. The number of forward moves from V_i to V_{i+1} is given by $\lceil \frac{\min(V_i, V_{i+1})}{\Delta} \rceil - \lceil \frac{\min(V_i, V_{i+1}, \dots, V_L)}{\Delta} \rceil + 1$.

From property 4, we see that the threshold may have to be tightened on the first visit to V_{i+1} if and only if

$\lceil \frac{\min(V_i, V_{i+1})}{\Delta} \rceil \Delta = \lceil \frac{V_i}{\Delta} \rceil \Delta$, i.e. if and only if this visit occurs on the first forward look from V_i when $T = \lceil \frac{V_i}{\Delta} \rceil \Delta$ and hence $V_i < T + \Delta$. This proves that the second subblock of Fig. 7, originally suggested by Gallager (3) as a simple means to implement the "first visit" test, is equivalent to the first. (Our proof of equivalence is strictly speaking only for trunk searches but it will soon be seen that it extends to general tree searches)

Property 6: (Superposition Property) If a forward move from V_i is made with threshold T , then the next backward move to V_i occurs with the same threshold T .

III. EXTENSIONS TO TREE SEARCHES

Trunk-searching property 6 is the key to understanding how the Fano algorithm searches a general value tree. Consider the simple dividing tree of Fig. 8 and suppose the algorithm terminates on V_L . Then from Fig. 2, in light of property 6, it follows that the trunk $V_0, V_1, \dots, V_L$ is searched as though the alternate path were not present since any advance down this alternate path eventually is followed by a return to V_i at the

departing threshold according to property 6. Moreover, the alternate trunk $V_0 V_1 \dots V_i V'_{i+1} \dots V'_L$ is also searched as though the path $V_{i+1} \dots V_L$ were not present, up to the point where the threshold at V_i is reduced to $\left[\frac{\min(V_i, V_{i+1}, \dots, V_L)}{\Delta} \right] \Delta$ for at this point there is no return to V_i from the move to V_{i+1} . The above comments, in light of property 6, can be seen to apply regardless of the number of different trunks in the value tree. We are thus led immediately to the following theorem which characterizes the final path found by the Fano algorithm:

Theorem 2: The final path $V_0, V_1, V_2, \dots, V_L$ through a value tree found by the Fano algorithm can be determined recursively as follows:

Let $P_{i+1}, P_{i+2}, \dots, P_L$ be a path from V_i through the tree. Then

$V_{i+1} = P_{i+1}^*$ where P_{i+1}^* is the best node value P_{i+1} for all those paths which maximize $\left[\frac{\min(V_i, P_{i+1}, \dots, P_L)}{\Delta} \right]$.

The number of forward looks made from V_i to V_{i+1} is exactly

$$\left[\frac{V_i}{\Delta} \right] - \left[\frac{\min(V_i, V_{i+1}, \dots, V_L)}{\Delta} \right] + 1.$$

The intuitive content of this theorem is that the Fano algorithm, at each node beginning with the origin, chooses as the final next node that node on the path through the tree stemming from the former node and having the greatest minimum node value. The following theorem is also an immediate consequence of our remarks above and characterizes the searching of non-final paths.

Theorem 3: Let $V_0, V_1, \dots, V_L$ be the final path through a value tree found by the Fano algorithm and let $P_{i+1}, P_{i+2}, \dots, P_{i+K}$ with the node P_{i+1} distinct from node V_{i+1} be a path of length K branching from V_i . Then P_{i+K} will be reached by the Fano search if and only if
$$\left[\frac{\min(V_i, P_{i+1}, \dots, P_{i+K})}{\Delta} \right] \geq \left[\frac{\min(V_i, V_{i+1}, \dots, V_L)}{\Delta} \right]$$

with the further proviso that node P_{i+1} be ordered better than V_{i+1} when equality holds. Moreover, when P_{i+K} is reached, the number of forward looks from P_{i+K} to the best node stemming therefrom is exactly

$$\left[\frac{P_{i+K}}{\Delta} \right] - \left[\frac{\min(V_i, V_{i+1}, \dots, V_L)}{\Delta} \right]$$

when V_{i+1} is ordered better than P_{i+1} , and is one greater than this otherwise.

References:

- (1) Wozencraft, J. M., "Sequential Decoding for Reliable Communication," IRE Convention Record, Part 2, pp. 11-25, 1957.
- (2) Fano, R. M., "A Heuristic Discussion of Probabilistic Decoding," IEEE Trans. on Info. Th., IT-9, pp. 64-74, April 1963.
- (3) Gallager, R. G., Private Communication, 1965.

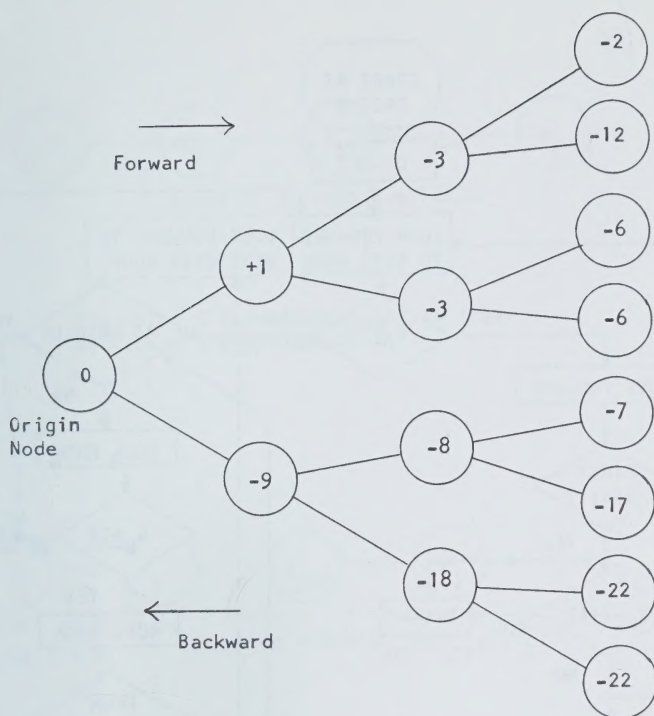


Figure 1. A Value Tree

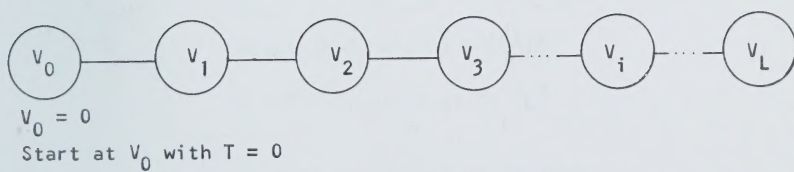


Figure 3. A General Value Trunk

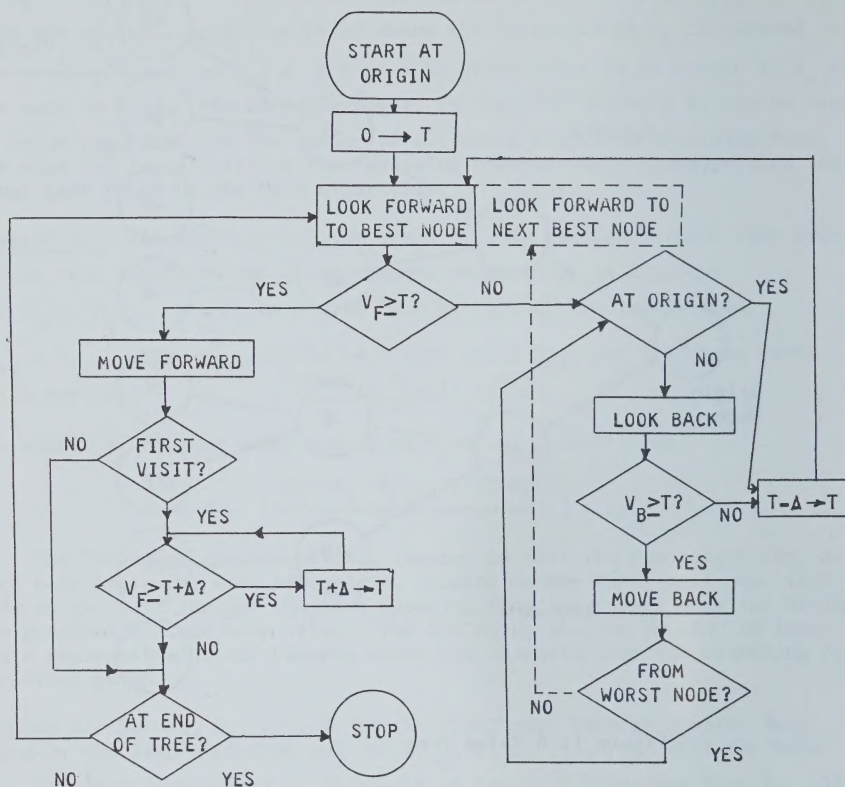


Figure 2. Flowchart of Fano Sequential Decoding Algorithm. V Is Value of Node Presently Occupied, V_B Is Value of Node One Branch Back in Tree, V_F Is Value of a Specified Node One Branch Forward in Tree.

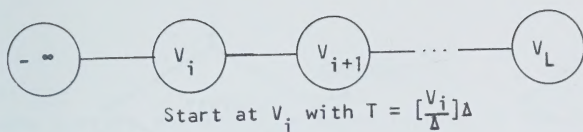


Figure 4. A Trunk Equivalent to That of Figure 3 Rightward of V_i .

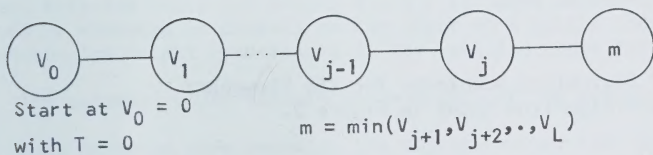


Figure 5. A Trunk Equivalent to That of Figure 3 Leftward of V_j .

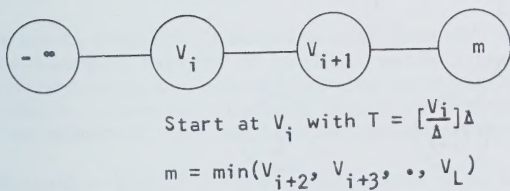


Figure 6. A Trunk Equivalent to That of Figure 3 between V_i and V_{i+1} .

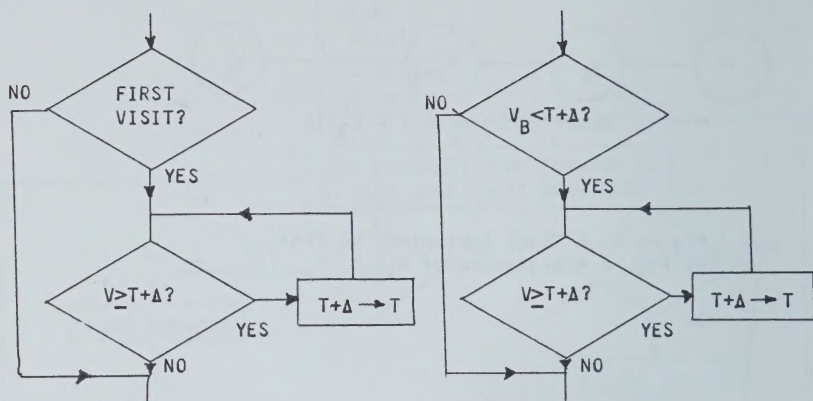


Figure 7. Equivalent Subblocks for the Flowchart of the Fano Algorithm Shown in Figure 2.

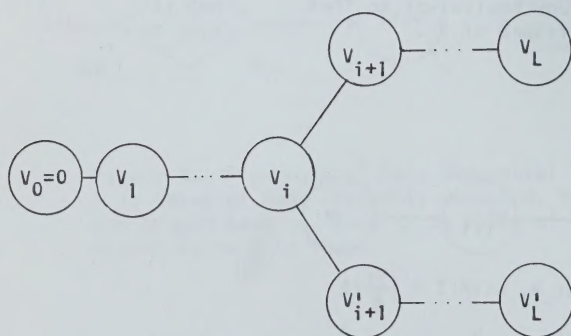


Figure 8. A Simple Dividing Value Tree

E. R. BERLEKAMP

Bell Telephone Laboratories, Incorporated, Murray Hill, New Jersey

ABSTRACT

This paper surveys the presently known results on the weight distributions of certain classes of binary codes.

INTRODUCTION

This talk summarizes presently known results on the weight distributions of certain binary codes, including the QR (quadratic residue) codes, the primitive BCH (Bose-Chaudhuri-Hocquenghem) codes, and the RM (Reed-Muller) codes. The audience is assumed to be familiar with the definitions of these codes, which may be found in Berlekamp's Algebraic Coding Theory (hereafter referred to as ACT).

Each of these three families of codes includes both cyclic codes (of length n) and extended codes (of length $N = n + 1$), each extended code being obtained by annexing an overall parity check to a cyclic code. It is easier to obtain the weight enumerators of the extended codes for two reasons:

- (1) The extended codes contain no codewords of odd weight.
- (2) Each extended code contains the all-one vector (of length N) so that A_1 , the number of codewords of weight 1, is given by $A_1 = A_{N-1}$ for all i .

It turns out that each of the extended codes of the three classes we consider is invariant under a transitive permutation group, and that the number of codewords of weight i in the cyclic code (denoted by a_i) may, therefore, be easily deduced from the weight distribution of the extended code via a straightforward theorem of Prange: [ACT p. 228]

$$a_{2j-1} = 2jA_{2j}/N = 2ja_{2j}/(N-2j)$$

The few known complete enumerations of extended QR, BCH, and RM codes rely on a considerable variety of results, which I have partitioned into seven different categories: code-preserving permutation groups, bounds on minimum distance, classical algebra, weight restrictions, duality, partial weight enumerators, and complete weight enumerators.

The implication relationships[†] among these results are shown in Fig. 1. For example, Kasami's weight restrictions for the 2nd order RM codes may be obtained either from the combination of McEliece's theorem with the Berlekamp-Sloane restrictions on RM codes of arbitrary order, or from the combination of the theorem that RM codes are invariant under the full affine group of an m -dimensional vector space over $GF(2)$ with Dickson's theorem on the binary canonical quadratic forms.

[†] The implication must be interpreted in the "practical" sense. I am not unaware of the fact that any true theorem logically implies any other.

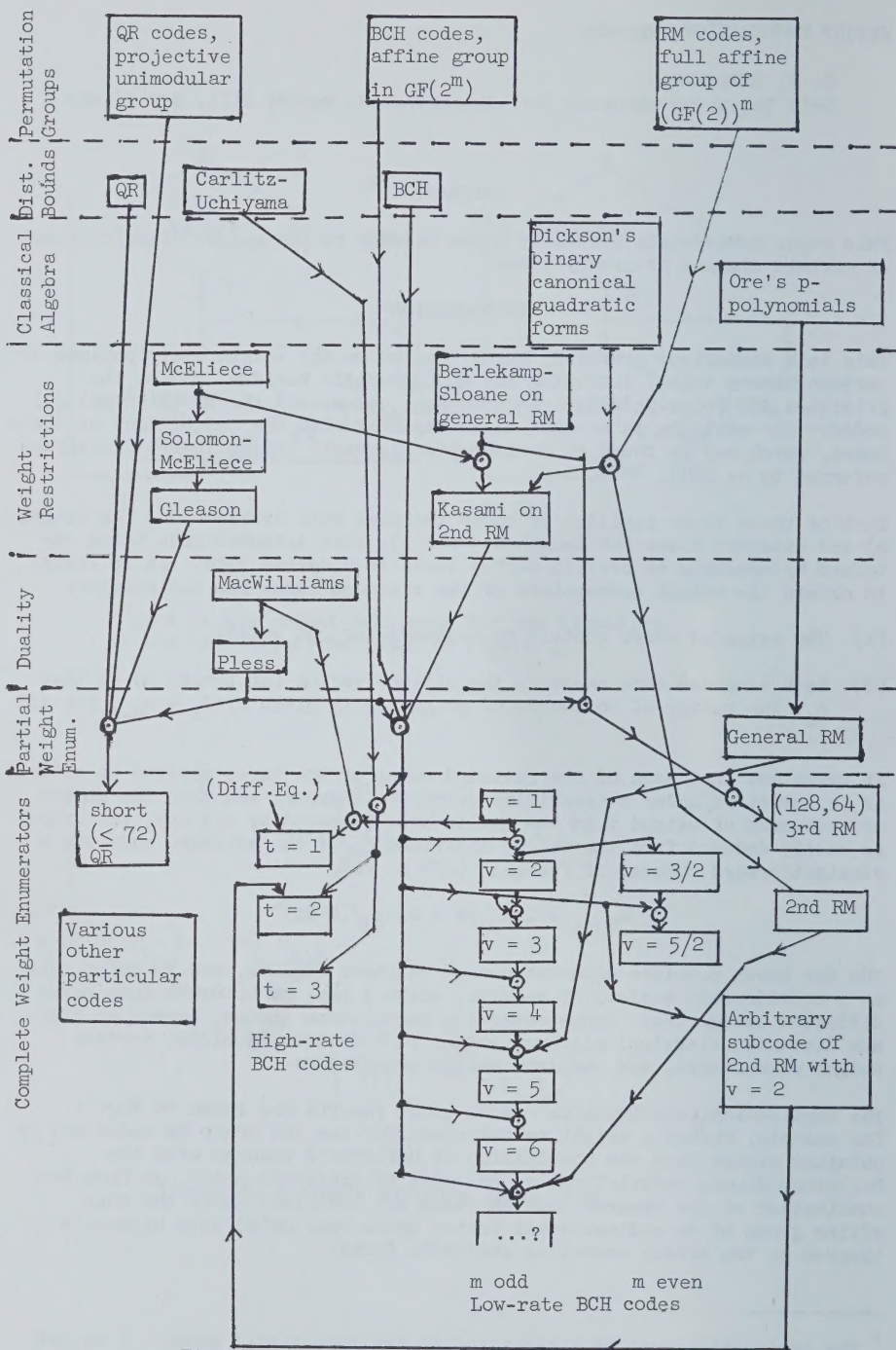


Figure 1. Implications Among Major Theorems

CODE-PRESERVING PERMUTATION GROUPS

A permutation group which preserves a code partitions the codewords of each weight into equivalence classes, each consisting of words which are permuted into each other by the permutations in the group. If the group has a large order, (i.e., comparable to the number of codewords in the code) then the number of classes may be sufficiently small that it is possible to obtain a weight enumeration of the codewords by counting the number in each equivalence class. Some of the groups which are known to preserve codes are the following:

Gleason, et al.; [ACT p. 357ff]

The extended QR code, with codewords coordinatized as $[C_0, C_1, \dots, C_{n-1}, C_\infty]$ (C_∞ denoting the overall parity check) is preserved by the projective unitary modular group, $C_i \rightarrow C_j$ where $j = \frac{ai+b}{ci+d}$ and a, b, c, d are any elements in $GF(n)$ such that $ad - bc = 1$.

Kasami, Peterson, Lin; [ACT p. 226f]

The coordinates of the codewords of an extended cyclic code of length $N = 2^m$ may be conveniently taken as elements of the finite field

$GF(N) = GF(2^m)$, a codeword then being $[C_{\alpha^0}, C_{\alpha^1}, C_{\alpha^2}, \dots, C_{\alpha^{-1}}, C_0]$.

($C_0 = C_{\alpha^0}$ denotes the overall parity check.) If α^j is a root of the generator polynomial, $j \neq 0$, and $\binom{k}{j} \neq 0$, then the code is preserved by the affine group on $GF(2^m)$: $C_\beta \rightarrow C_\alpha$ where $\beta = \xi\alpha + \delta$ for any $\delta \in GF(N)$, $\xi \in GF(N) - 0$.

The extended primitive BCH codes satisfy the hypotheses of this theorem.

The binomial coefficient $\binom{k}{j}$ is easily evaluated in $GF(2)$ with the aid of a theorem of Lucas (1878) [ACT p. 113]. If every digit of the binary expansion of the integer k is at least as great as the corresponding digit of the binary expansion of the integer j , then j is said to be a descendant of k and $\binom{k}{j}$ is odd. If j is not a descendant of k , then $\binom{k}{j} = 0$ in $GF(2)$.

The permutations of the preceding theorem may be used to obtain relationships between the weight enumerators of several different codes. Let $A^{(\bar{K}, a, b)}$ be the extended primitive cyclic code whose generator polynomial's roots are α^a, α^b , and their conjugates, as well as α^k for all k in the set $\bar{K}$. Let $A^{(\bar{K}, a, b)}$ be the real polynomial whose i th coefficient is the number of codewords of weight i in $A^{(\bar{K}, a, b)}$.

Theorem

If $\bar{K}$ contains all of its own descendants, and if b is the unique descendant of a not in $\bar{K}$, then

$$A^{(\bar{K})} - A^{(\bar{K}, b)} = {}_2 \deg(\alpha^a) \left(A^{(\bar{K}, a)} - A^{(\bar{K}, a, b)} \right)$$

The proof follows immediately from the observation that the translations of the affine group induce a $2^{\deg(\alpha^8)}$ -to-one correspondence between the code-words in the corresponding codes.

For example, if $N > 32$, then we may obtain a formula which gives the weight enumerator of the 4-error-correcting extended BCH code in terms of the enumerators of a non-BCH code and the 2- and 3-error-correcting extended BCH codes:

$$A^{(1,3,5,7)} = A^{(1,3,7)} - \frac{A^{(1,3)} A^{(1,3,5)}}{N}$$

The affine group on $GF(2^m)$ is only a subgroup of the complete permutation group which preserves RM codes, as given by the following theorem:

RM Invariance [ACT p. 366]

The coordinates of the codewords of the Reed-Muller code of length 2^m may be conveniently taken as m -dimensional binary vectors. The RM code is preserved by the complete affine group over $GF(2)$: $C \rightarrow C$ where

$\vec{y} = A\vec{x} + \vec{b}$, for any invertible $m \times m$ binary matrix A and any m -dimensional binary vector $\vec{b}$.

Some codes, including all nontrivial RM codes, are preserved only by permutations in one of the above groups, but other codes are preserved by much larger groups, of which the above are proper subgroups. For example, the extended binary QR code of length 24 is preserved under the complete Mathieu group of order $16 \cdot 15 \cdot 14 \cdot 12 \cdot 8 \cdot 23 \cdot 11 \cdot 3$. [Assmus, Mattson (1964); ACT p. 395].

No other nontrivial extended binary cyclic code which is preserved by a transitive permutation group is known. It is known that none exist of length ≤ 45 . [ACT p. 432-433].

However, there are other classes of binary codes with large permutation groups. For example, Prange has argued that PG codes are preserved by the collineation groups of the corresponding projective planes, but this result has not yet appeared in the literature. Nor has it been used to obtain weight enumerators of binary PG codes longer than 73.

The complete permutation groups which preserve most codes are not known.

BOUNDS ON MINIMUM DISTANCE

Mattson-Solomon (1961), [ACT p. 353ff]

The minimum odd weight of the augmented cyclic binary QR code of length n is at least

$$d \geq \begin{cases} \sqrt{n} & \text{if } n \equiv 1 \pmod{8} \\ (1 + \sqrt{4n-3})/2 & \text{if } n \equiv -1 \pmod{8} \end{cases}$$

Carlitz-Uchiyama (1957)

The minimum weight of the dual of the extended t-error-correcting binary BCH code of length m is at least

$$d \geq 2^{m-1} - (t-1)2^{m/2}$$

This result follows immediately from a paper by Carlitz-Uchiyama (1957). Although the underlying result was apparently conjectured as early as 1935 and proved (as a consequence of Weil's deep proof of the Riemann hypothesis for function fields) in 1941, its application to coding was first noticed by Anderson (1968). Unfortunately, no elementary proof of the Carlitz-Uchiyama bound for the minimum weight of the dual of the BCH code is known. The theorem is quite strong; it guarantees that the duals of high-rate BCH codes generally have considerably higher distances than BCH codes of the same low rate.

The BCH bound asserts that the minimum distance of the t-error-correcting BCH code is at least $2t - 1$.

Many proofs are known. The first were Hocquenghem (1959) and Bose-Chaudhuri (1960); the "most elegant" is due to Mattson-Solomon (1961) [ACT p. 408]; the most constructive (from the standpoint of decoding) appears in ACT, Chapters 7 and 10.

The QR bound is known to be slack in many cases [ACT p. 360]. The BCH bound is known to be slack for certain nonprimitive BCH codes (of length $\neq 2^m - 1$), [ACT p. 409f], but it is conjectured that the BCH bound gives the exact minimum distance of all primitive BCH codes (of length $2^m - 1$). Many special cases of this conjecture have been proved [ACT pp. 265-270]. The Carlitz-Uchiyama bound is conjectured to be tight when m is even, but there are cases with odd m when it is known to be slack.

RESULTS FROM CLASSICAL ALGEBRA

Dickson (1900) [ACT p. 414]

Any binary quadratic form in m variables $F = \sum_{i=1}^m \sum_{j=1}^m B_{i,j} x_i x_j$ can be reduced by an invertible linear transformation of the x 's to one of the canonical forms

$$F = \begin{cases} x_k^2 + \sum_{i=1}^{(k-1)/2} x_{2i-1} x_{2i} & k \text{ odd} \\ \lambda \left(x_k^2 + x_{k-1}^2 \right) + \sum_{i=1}^{k/2} x_{2i-1} x_{2i} & k \text{ even, } \lambda = 0 \text{ or } 1 \end{cases}$$

Ore (1932-1933), [ACT p. 248]

A polynomial has the form

$$A(x) = -u + \sum_i L_i x^{2^i}, \quad u, L_i \in GF(2^m), \quad L_0 \neq 0$$

iff its roots (in $GF(2^{\infty})$) form an affine subspace over $GF(2)$.

WEIGHT RESTRICTIONS

McEliece (1967)

If the check polynomial of a cyclic code does not have any j -tuple of roots,

$$\beta_1, \beta_2, \dots, \beta_j, \text{ such that } \prod_{i=1}^j \beta_i = 1,$$

then the weight of every codeword in that code is a multiple of 2^j .

The special case $j = 1$ is trivial, for it merely states that if $(x-1)$ divides the generator polynomial, then all weights are even. The special case $j = 2$ is due to Solomon-McEliece (1966), [ACT p. 411]. The special case of the Solomon-McEliece theorem applied to quadratic residue codes of length $\equiv -1 \pmod 8$ was previously proved by Gleason. The proof of the general McEliece theorem follows the same general lines as the Solomon-McEliece theorem, using the Γ_j machinery first introduced by Solomon (1962). This is a powerful tool which may yet yield still further results.

The McEliece theorem guarantees that all weights in the r^{th} order RM code of length 2^m are divisible by $2^{\lceil (m-1)/r \rceil}$.

Berlekamp-Sloane (1968)

The only weights $< 2d$ in a Reed-Muller code of minimum distance d are all of the form $2d - 2^i$ for some i .

This theorem has also been proved for many supercodes of the RM codes.

In the special case of 2nd order RM codes, $d = N/4$, so Berlekamp-Sloane assert that every weight must be of the form $N/2$ or $N/2 \pm 2^i$. McEliece guarantees that $i \geq \lceil (m-1)/2 \rceil$. The result was first obtained by Kasami via other arguments:

Kasami [ACT, p. 413]

Every weight in any subcode of the 2nd order RM code of length 2^m is of the form

$$2^{m-1} + \epsilon 2^{(m+i)/2-1}$$

where i is some nonnegative integer $\equiv \pmod 2$ and $\epsilon = 0, +1, \text{ or } -1$.

DUALITY

MacWilliams [ACT p. 400f]

Let $\mathcal{Q}$ be a linear binary code of dimension k , with weight enumerator $A(z) = \sum_i A_i z^i$, and let $\mathcal{Q}^\perp$ be its dual code, with weight enumerator $B(z) = \sum_i B_i z^i$. Then

$$A(z) = z^k \left(\frac{1+z}{2} \right)^N \overline{B} \left(\frac{1-z}{1+z} \right)$$

The MacWilliams identities enable one to determine the weight enumerator of any code from the weight enumerator of its dual code.

A transformation of the MacWilliams identities yields the Pless identities, which have the advantage that it is possible to determine the enumerators of both $\mathcal{Q}$ and $\mathcal{Q}^\perp$ from sufficient partial information about each.

Pless [ACT p. 405]

$$\sum_{j=0}^N (N-2j)^r A_j = z^k \sum_{j=0}^r B_j F_r^{(j)}(N) \quad \text{for all } r$$

where

$$F_r^{(j)}(N) = \frac{d^r}{dx^r} [\cosh^{N-j} x \sinh^j x]_{x=0}$$

In particular, if $B_0, B_1, B_2, \dots, B_{s-1}$ are known and only s A's are unknown, the Pless identities may be solved for all the remaining A's and B's. It turns out that the Pless equations are identical to the equations one solves for the error values of a nonbinary BCH code after the error locations have been found.

PARTIAL WEIGHT ENUMERATORS

Although the complete weight enumerators for arbitrary RM codes are not yet known, the following partial information is available:

Theorem [ACT p. 363f]

The number of codewords of minimum weight in the r th order RM code of length 2^m is given by

$$A_{2^{m-r}} = \frac{2^m (2^m - 2^0) (2^m - 2^1) (2^m - 2^2) \dots (2^m - 2^{m-r-1})}{(2^{m-r})(2^{m-r} - 2^0)(2^{m-r} - 2^1)(2^{m-r} - 2^2) \dots (2^{m-r} - 2^{m-r-1})}.$$

This result is obtained by geometric arguments which identify each minimum-weight codeword with the indicator function of an $(m-r)$ -dimensional affine subspace of $(\text{GF}(2))^m$ over $\text{GF}(2)$.

By similar arguments, one might obtain the number of words of minimum weight in PG and EG codes.

The recent results of Berlekamp-Sloane offer hope that a count of RM code-words of each weight less than twice the minimum may soon be forthcoming, but such a result has not yet been obtained.

COMPLETE WEIGHT ENUMERATORS

Using her identities, Pless (1963) succeeded in determining the complete weight distribution of certain binary QR codes with lengths up to 72.

Using known permutation groups which leave all cyclic binary codes invariant, MacWilliams, Goethals, Kasami-Lin-Peterson, Golomb-Welch, and McEliece have undertaken computer searches to determine the weight distributions of many cyclic binary codes and extended codes of several lengths. In fact, most of the theorems surveyed in this paper were first conjectured on the basis of computer-generated results. Successful conjectures are most often obtained from inspection of the factors of the numbers of codewords of each weight. The numbers themselves are often enormous integers which cannot be readily interpreted except in factored form.

The most extensive (and impressive) collection of weight distribution formulas yet to appear were first discovered by Kasami (1968). It was he who first discovered that the BCH bound and his restrictions on the weights of the 2nd order RM code were sufficient to solve the Pless identities for the weight distributions of many extended BCH codes which are subcodes of the second order RM code of length 2^m . In particular, for odd m , Kasami's attack succeeded for 1-, 2-, and 3-error-correcting extended BCH codes, as well as for the low-rate BCH codes having $k = vm + 1$ information symbols, with $v = 1, 2, 3$, or 4. Using special arguments, Kasami also succeeded in obtaining complete enumerators of arbitrary subcodes of the 2nd order RM code having $k = vm + 1$ information symbols, with $v = 3/2$ or 2. This gave him the enumerator of the 2-error-correcting BCH codes with even m , and the low-rate BCH codes with $v = 3/2$. Using the enumerator of the $v = 3/2$ codes and the Pless identities yielded the results for $v = 5/2$. Kasami also conjectured an enumerator of the 3-error-correcting BCH codes with even m , but he succeeded in proving this only for $m < 12$. I recently noticed that the additional restriction imposed by the Carlitz-Uchiyama bound is sufficient to prove this conjectured enumerator. It also proves the enumerator for even m , $t = 2$, without any special arguments.

Berlekamp (1968) observed that the known partial weight enumerator for the general RM code provides an additional constraint sufficient to extend Kasami's odd- m , low-rate BCH (RM subcode) enumerations to the cases $v = 5$ and $v = 6$. Using the recently calculated complete weight enumerators for the 2nd order RM codes, it now appears possible to extend Kasami's odd- m , low-rate BCH (RM subcode) enumerators to even more cases, $v = 7, \dots$. However, this calculation has not yet been carried out, and there is no guarantee of success because the success of the calculation for $v = 2, 3, 4, 5$, and 6 depends on certain apparently fortuitous results which may not recur.

Berlekamp-Sloane (1968) have shown that the weight enumerator for the 2nd order RM code of length 2^m is

$$A_{2^m \pm 2^{m-1-j}} = 2^j(j+1) \left\{ \frac{(2^m-1)(2^{m-1}-1)}{4-1} \right\} \left\{ \frac{(2^{m-2}-1)(2^{m-3}-1)}{4^2-1} \right\} \dots$$

$$\left\{ \frac{(2^{m-2j+2}-1)(2^{m-2j+1}-1)}{4^j-1} \right\}$$

$$A_{2^{m-1}} = 2^{m^2+m+1} - 2 \sum_{j=0}^m A_{2^{m-2j}}.$$

This result was obtained by counting the number of quadratic forms at each stage of the proof of Dickson's theorem.

The known restrictions on the 3rd order RM code of length 128 are now sufficient to determine this code's weight enumerator from the Pless identities, but this calculation, requiring multiprecision or mixed residue arithmetic, has not yet been programmed. If the results should yield evidence of any new weight restrictions for 3rd order RM codes, it might facilitate enumerations of certain BCH codes which are subcodes of 3rd order RM codes.

More generally, what are the possible solutions (with nonnegative weights) of the Pless identities for codes which are equivalent to their own duals and which have only weights which are multiples of 4 and greater than some prescribed minimum? (The extended QR codes of lengths $\equiv 0 \pmod 8$ and the $(m-1)/2$ order RM code of length 2^m are examples of such codes.)

What is the geometric interpretation of the low weight codewords in the low-rate BCH subcodes of the 2nd RM code (odd m)?

Are there results comparable to the MacWilliams-Pless theorems for Lee-metric weight enumerators of dual codes?

What is the enumerator of coset leaders for various "simple" codes, such as the first and second order RM codes and their BCH subcodes?

In my opinion, these unsolved questions are now ripe for further research.

REFERENCES

1. D. R. Anderson (1968), "A New Class of Cyclic Codes", J. SIAM **16** pp. 181-197.
2. E. R. Berlekamp (1968), Algebraic Coding Theory, McGraw-Hill.
3. E. R. Berlekamp and N. J. A. Sloane (1968), paper to appear.
4. L. Carlitz and S. Uchiyama (1957), "Bounds for Exponential Sums", Duke Math. J. **24**, pp. 37-41.
5. R. J. McEliece (1967), "Linear Recurring Sequences Over Finite Fields", Ph.D. dissertation, Mathematics Department, California Institute of Technology.

6. A. Weil (1941), "On the Riemann Hypothesis in Function Fields", Proc. Nat. Ac. of Sci. 27, pp. 204-207.

(A more complete list of references appears in Algebraic Coding Theory.)

DJIMITRI WIGGERT

Department of Electrical Engineering
University of Massachusetts
Amherst, Massachusetts

ABSTRACT

An easily-implemented method is described for detecting threshold decoding errors based on the behavior in time of the sum of the parity checks orthogonal on the digit to be decoded. A sequence of matrices, giving successive values of these orthogonal parity checks, is developed to facilitate studying this behavior. The parameters of interest are the peak value of the sum and the length of time required to achieve this value and return to zero. The chief application of this technique is in sensing noise bursts while decoding in a random-error-correction mode.

I. INTRODUCTION

Because of their ease of implementation and because their error-control properties compare favorably with those of block codes, convolutional (or recurrent) codes have been used for error protection in several practical digital data transmission systems [1,2]. Most decoding algorithms which use convolutional codes suffer from the possibility of a serious (though usually not catastrophic) mode of failure - error propagation. This is the tendency of an initial decoding error to trigger a succession of further errors even though correct reception of transmitted digits may have resumed.

A number of schemes have been devised to cope with this problem. Perhaps the earliest were two proposed by Wozencraft and Reiffen [3] for their form of sequential decoding. The technique of periodic resynchronization simply forces any error propagation then in progress to terminate by "starting from scratch" at regular intervals (usually several thousand digits in length). A price is paid in terms of a slight decrease in overall information rate. The second technique is to exploit, by sending a retransmission request, the occurrence of a sequential decoding error, as signaled by a sharp increase in the computational load at the decoder. Robinson [4] has proposed a scheme, definite decoding, in which there is no feedback of decoding decisions (based on syndromes) within the decoder. As will be shown below, such feedback is helpful as long as decoding is correct, but it is the cause of propagation of what might otherwise have been as little as a single decoding error. Definite decoding succeeds at the price of increased probability of the initial decoding error. Sullivan [5] has suggested yet another solution, based on the fact that the all-zero syndrome is more probable than any other one: when the decoder is suspected of being in trouble, reset the entire syndrome to zero and continue decoding from there. Finally, encouraging results have been obtained [4,6] in the construction of codes which have the property of limiting the error propagation due to a decoding error.

In this paper, we consider one way in which it may be determined whether the decoder has committed an error. This technique is most effective for bursts of channel noise and applies specifically to a threshold decoder operating in a random-error-correcting mode. In fact, with some additional storage and logic, it could be used to prevent a decoding error by causing the decoder to enter a burst-correcting mode and re-process the

portion of the received data stream which would have produced the incorrect output.

II. BACKGROUND

A. Encoding and decoding

Following Robinson and Bernstein [6], we first note that in the general case for binary convolutional codes, the semi-infinite parity-check matrix H has the form of Figure 1 for a code of rate $R=k/(k+r)$ and actual constraint length $n_A = (k+r)(m+1)$. Here I_r is the $r \times r$ identity matrix, and B is shown in detail in Figure 2. The elements of B are given using Massey's notation [2]. All of the unshaded portions of H contain only 0's. Then for encoding the equation

$$H \underline{x} = \phi \quad (1)$$

must be satisfied, where $\underline{x}$ and ϕ semi-infinite column vectors, the latter consisting entirely of 0's. The operation described by eq. (1) is easily seen to be the following: A 1 on the main diagonal of a submatrix I_r picks out a (check) digit in $\underline{x}$ as the linear combination of rk information digits ($1 \leq l \leq m+1$) checked by the elements of the l B-blocks in the same row. Thus, the transmitted code vector $\underline{x}$ consists of alternating runs of k information digits and r check digits.

It is clear that an easy visualization of the "transient" and "steady-state" behavior of a convolutional code is possible by considering only the finite matrix H_N , made up of the first $(m+1)r$ rows and $(m+1)(k+r)$ columns of H . More important is the fact that the decoding behavior is specified completely (in terms of error-correcting properties) by H_N , since (as we shall soon see) a decoding decision will be based on certain functions of the most recent n_A received digits.

For the case in which $k = r = 1$, we have $R = 1/2$, $n_A = 2(m+1)$. Then B and $\underline{x}$ reduce to

$$B = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_m \end{bmatrix} \quad \underline{x} = \begin{bmatrix} i_0 \\ c_0 \\ i_1 \\ c_1 \\ \vdots \end{bmatrix}$$

where the symbols i and c represent information and check digits, respectively. Since this gain in simplicity of the code can be accomplished without loss of generality, we shall consider only rate one-half codes from this point on. Figure 3 shows an encoder for such a code.

As an example, consider Massey's "trail-and-error" code [2] having $R = 1/2$ and $m = 5$. Then H_N is enclosed by the dashed lines in

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad H = \begin{bmatrix} 11 & 11 & 11 & 11 & 11 & 11 \\ 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 11 & 11 & 11 & 11 \\ 10 & 00 & 00 & 11 & 11 & 11 \\ 10 & 10 & 00 & 00 & 11 & 11 \\ 10 & 10 & 10 & 00 & 00 & 11 \\ \hline & 10 & 10 & 10 & 00 & 11 \\ & & 10 & 10 & 00 & 00 \\ & & & 10 & 10 & 00 \\ & & & & 10 & 10 \\ & & & & & 10 \\ & & & & & & 10 \end{bmatrix}$$

We shall assume the usual additive-noise channel which adds error digits (1's) at random to the transmitted digits. Thus, if $\underline{e}$ and $\underline{y}$ denote the (semi-infinite) error and received vectors, respectively, we have

$$\underline{y} = \underline{x} + \underline{e}$$

Decoding then takes place based on the syndrome $\underline{s}$, also a column vector, defined just as in block coding by

$$\underline{s} = H\underline{y} = H\underline{e}$$

in view of equation (1). A decoder which operates in this manner is shown in Figure 4. Received information digits i_j enter a replica of the encoder. Any time a particular parity-check relation is satisfied, a 0 is fed to the syndrome shift register (SSR); otherwise a 1 is fed. Each time a new syndrome digit enters the SSR, the oldest digit is "bumped" at the other end and the decoding function $f(S)^{\dagger}$ computes an estimate e_j^* of the error digit component e_j of the information digit i_j about to leave the encoder. If $e_j = 1$, then it has affected, by its presence as i_j passed through the encoder circuitry, some subset of the digits now in the SSR; the effect of e_j should be removed, and it will be if $e_j^* = e_j$. This type of decoder operation is known as feedback decoding. If $e_j^* \neq e_j$, the result, unfortunately, is not merely a decoding error on i_j ; the incorrect information fed back into the SSR generally causes additional decoding errors on subsequent information digits (which might otherwise have been decoded correctly), and we have the phenomenon of error propagation.

B. Threshold decoding

In the simplest form of threshold decoding, the function $f(S)$ applies a majority decision rule to certain linear combinations of the digits in the SSR. Since $\underline{s} = H\underline{e}$, each digit of $\underline{s}$ checks certain positions of the error vector. The key step here is to form a set $\{A_i\}$ of linear combinations of digits in the SSR such that each A_i checks e_j , the position about to be decoded, and none of the other $(m+1)(k+r)-1$ e 's checked by the digits presently in the SSR is checked more than once by the entire $\{A_i\}$. Such a set of parity checks is said to be orthogonal on e_j . It can be shown [2] that if $\{A_i\}$ has J elements, then the majority decoding rule is: choose $e_j^* = 1$ if more than $\lfloor J/2 \rfloor$ of the A_i 's are 1; otherwise choose $e_j^* = 0$. (The symbol $\lfloor x \rfloor$ denotes the largest integer equal to or less than x , i.e., the integer part of x . For future use, we

$\dagger$ We use lower- and upper-case S to denote the semi-infinite syndrome vector and the contents of the SSR, respectively.

also introduce the symbol $[x]$ to denote the smallest integer equal to or greater than x . This notation is due to Massey [2].) Thus, the decoding operation reduces to comparing $\sum_{i=1}^J A_i$ (as the usual arithmetic, rather than mod 2, sum) with the threshold value $[J/2]$.

As an example of the construction of orthogonal parity checks, a set of $J=4$ parity checks orthogonal on e_j can be formed for the code given in section II-A by taking the first, fourth, fifth, and second plus sixth rows of H . This gives a $J \times (m+1)(k+r)$ matrix which we shall call A . In this case

$$A = \begin{bmatrix} 11 & 00 & 00 & 00 & 00 & 00 \\ 10 & 00 & 00 & 11 & 00 & 00 \\ 10 & 10 & 00 & 00 & 11 & 00 \\ 10 & 01 & 10 & 00 & 00 & 11 \end{bmatrix} \quad (2)$$

Column 1 of A checks e_j , column 2 checks e_{j+1} , etc. The orthogonality property is apparent from a glance at A . We shall say that a matrix is orthogonal on e_j if its rows define a set of parity checks orthogonal on e_j .

III. ANALYSIS OF $\{A_i\}$ AS A FUNCTION OF TIME

In order to visualize the time-behavior of the decoder, we begin by representing contents of the SSR at successive time intervals in a manner used by Robinson [4]. This involves nothing more than first writing down the column vector $\underline{s}$ and then examining components s_0 to s_m , s_1 to s_{m+1} , s_2 to s_{m+2} , etc., as successive values of $\underline{s}$. This procedure must be accompanied by modifications in $\underline{s}$ whenever $\underline{e}^* = 1$. Table I illustrates the technique for the code of Section II, with $\underline{e}^T = (0000000000110...)$ where the error pattern begins no earlier than e_{11} to permit complete examination of its effect on the behavior of ΣA_i . We note also that the columns of any B and its adjacent I_r can be used to determine $\underline{s}$ for any given error pattern. This is because a component of $\underline{s}$ is nothing more than the sum of the columns of H having the same number as the components of $\underline{e}$ which are 1's. Thus, calculating $\underline{s}$ for an error pattern which is checked entirely by a single BI_r is a trivial matter. The positions of the corresponding components of $\underline{s}$ will be the row numbers (with respect to H) of the BI_r used. For an error pattern which overlaps two or more BI_r 's, "phase shifting" may be applied to obtain $\underline{s}$ based on components calculated for the single- BI_r case. For example, for the code of Section II,

$\underline{e}_1^T = (100...)$ yields $\underline{s}_1^T = (10011100...)$ having its first six components equal to the single column of B , while $\underline{e}_2^T = (0100...)$ yields $\underline{s}_2^T = (1000000...)$, corresponding to the identity matrix I_1 . Thus, $\underline{e}_3^T = (1010000...)$ will give $\underline{s}_3^T = (1001110...)+(01001110...)=(11010010...)$, while $\underline{e}_1 + \underline{e}_2$ will give $\underline{s} = \underline{s}_1 + \underline{s}_2$.

The same approach can be applied to calculating $\{A_i\}$ for successive states of the decoder. To accomplish this, let $A^{(1)}$ be A as defined by equation (2). Then let $A^{(2)}$ have its rows given by rows 2, 5, 6, and 3 plus 7 of H . Continuing in this manner, we terminate with $A^{(6)} = A^{(m+1)}$ having its rows equal to rows 6, 9, 10, and 7 plus 11 of H . It will be noted that $A^{(1)}$ has 2 (in general, $k+r$) more columns than $A^{(i-1)}$. Furthermore, if $A^{(1)}$ is orthogonal on e_j , then $A^{(2)}$ is orthogonal on

Table I.

Behavior of contents of syndrome register as a function of time. (Intersections of starred rows and columns are entries which were changed as a result of feedback.)

$$\underline{e}^T = (0000000000110\dots)$$

$$\Sigma A_i = 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 0 \ 0 \ \dots$$

<u>S</u>	*
0	0
0	0 0
0	0 0 0
0	0 0 0 0
0	0 0 0 0 0
0	0 0 0 0 0 0
0	0 0 0 0 0 0
0	0 0 0 0 0 0
*1	1 1 1 0
*1	1 1 0 etc.
*1	1 0
0	0
:	
.	

$e_{i+k+r} = e_{j+2}$, etc. Now, from Table I we observe that removing the effect of $e_{11} = 1$ is accomplished by adding the transpose of (100111) to the contents of the SSR as shown in column 6 of the table. As far as future values of S are concerned, this is the same as changing e_{11} to zero. In general, $e_i^* = 1$ implies effectively reversing the binary value of e_i for all future decoding operations. This observation carries over unchanged to the analysis of the time behavior of ΣA_i via matrices $A^{(1)}$ through $A^{(m+1)}$. We can, therefore, dispense with H and Table I, once these matrices have been calculated. We note, in addition, that we can define matrices $A^{(i)}$ for $i > m+1$; these differ among themselves and from $A^{(m+1)}$ only in having the form $[\Phi A^{(m+1)}]$, where Φ is an all-zero matrix. Therefore, the same analysis can be accomplished by advancing the components of $\underline{e}$, $(k+r)$ positions at a time toward the e_1 position and continuing to multiply by $A^{(m+1)}$ the succession of $\underline{e}$'s produced in this way.

Continuing with the example of this section, we find

$$A^{(2)} = \begin{bmatrix} 00 & 11 & 00 & 00 & 00 & 00 & 00 \\ 10 & 10 & 00 & 00 & 11 & 00 & 00 \\ 10 & 10 & 10 & 00 & 00 & 11 & 00 \\ 00 & 10 & 01 & 10 & 00 & 00 & 11 \end{bmatrix}$$

and finally

$$A^{(6)} = \begin{bmatrix} 10 & 10 & 10 & 00 & 00 & 11 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 10 & 10 & 10 & 00 & 00 & 11 & 00 & 00 \\ 00 & 00 & 00 & 00 & 10 & 10 & 10 & 00 & 00 & 11 & 00 \\ 00 & 10 & 10 & 10 & 00 & 10 & 01 & 10 & 00 & 00 & 11 \end{bmatrix}$$

In fact, since $A^{(m+1)}$ has dimensions $J \times (2m+1)(k+r)$, it alone could be calculated and used to perform by inspection the entire analysis in the manner just indicated [i.e., sliding the error vector past a "window" of length $(2m+1)(k+r)$]. For $\underline{e}$ as before, we get for $A^{(1)}\underline{e}$, $A^{(2)}\underline{e}$, ..., $A^{(6)}\underline{e}$, $A^{(7)}\underline{e}$ the vectors

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

where e_1 differs from e in having e_{11} , the error digit estimated from $A^{(6)}e$, changed from 1 to 0 in view of the fact that $e_{11}^* = 1$.

IV. UTILIZATION OF ΣA_i TO DETECT DECODING ERROR

With the method developed in section III, we can easily study the behavior of ΣA_i for various error patterns. As an illustration of this behavior as a function of time for a correctable and an uncorrectable pattern, consider the example just concluded. From the theory of threshold decoding [2] we know that this code will correct any pattern containing up to $\lfloor J/2 \rfloor = 2$ errors out of $n_A = 12$ successive digits. (Certain patterns with more than $\lfloor J/2 \rfloor$ errors are, in general, also correctable by any code.) Thus e of the example is correctable. The aspect to be noted is the sequence of values assumed by ΣA_i : 0, 1, 2, 3, 0. Clearly, whenever the decoder estimates a 1 for an information digit error, $\lfloor J/2 \rfloor < \Sigma A_i < J$. As a correctable pattern works its way through the decoder, the behavior of ΣA_i consists therefore of a series of values starting from 0 (assuming no channel errors in the decoder) and increasing to at most $\lfloor J/2 \rfloor$ until the first information error reaches the last (or e_0) stage of the encoder shift register. Then ΣA_i becomes and remains greater than $\lfloor J/2 \rfloor$ until all information errors have been corrected. For high-density bursts, the return to $\Sigma A_i = 0$ is rapid compared with the build-up just described. There are two reasons for this. First, as pointed out earlier, the effect of feeding back $e_i^* = e_i = 1$ is to change e_i to 0 as it is seen by the various parity-check equations. This will, in general, cause more of these equations to be satisfied (i.e., sum to 0). Second, for the code being used, $J=4$ while $n_A = 12$. Thus a correctable pattern for this code will contain at most 2 errors. If these are closely bunched (i.e., burst length b small compared to n_A), then the burst, moving at $k+r = 2$ positions per time unit through the decoder is processed in 2 or 3 additional time units at most once its leading digit reaches the last stage of the encoder. It can also be seen that after returning to zero, ΣA_i must remain there for, say, $m+1$ time units. (This quantity would vary with the code.) This is because some correctable error patterns give sequences of ΣA_i values which include one or more consecutive zeroes.

Thus there are two properties of ΣA_i to be observed in detecting a decoding error on a burst. The first is the sequence of values taken on. The second is the length of time to reach a peak value and the length of time to return to 0. We have seen that for a correctable burst of length b having at least one information-digit error, ΣA_i should go from 0 to the range $\lfloor J/2 \rfloor$ to J in about $m+1$ time units or less and should return to 0 in $\lfloor b/(k+r) \rfloor$ time units and remain there. For a correctable burst having only check-digit errors, $\Sigma A_i < \lfloor J/2 \rfloor$ all the while the burst is being processed. The length of time that $\Sigma A_i \neq 0$ is the same as in the previous case, no more than $\lfloor m+b/(k+r) \rfloor$ time units.

Now consider behavior for an uncorrectable burst. If the burst leads to error propagation, then of course ΣA_i may be a long time in returning to 0. This is a clear distinction from behavior for correctable bursts, and it can, in fact be used easily to prevent propagation of a decoding error by causing the decoder to enter a burst-correcting mode in conjunction with a delay in outputting decoding decisions i^* on the offending portion of the digit stream while the burst mode is re-processing it. If the burst results in decoding errors but no error propagation, no useful information is

obtained from the sequence of values assumed by ΣA_i . The length of time for which $\Sigma A_i \neq 0$ still supplies the indication that all is not well. In view of incorrect feedback from the SSR, the effect of received information digit errors is not removed, thus permitting these to be "seen" by columns of $A(m+1)$ to the left of the single all -1's column which checks e_j just prior to the decoding decision e_j . This causes ΣA_i to remain non-zero for more than $\lceil m+b/(k+r) \rceil$ time units, and this fact can be used to enter a burst-correcting mode as before. For example, for the code considered earlier, the error pattern 111 occurring at i_j c_j i_{j+1} gives the sequence of values $\Sigma A_i = 1, 1, 2, 1, 2, 2, 2, 0, 1, 0, \dots$, which is of length 10 compared with $\lceil m+b/(k+r) \rceil = 7$. Here there are decoding errors on the second and third occurrences of $\Sigma A_i = 2$.

The method is not foolproof. For example, the pattern 10101 occurring as c_j i_{j+1} c_{j+1} i_{j+2} c_{j+2} gives $\Sigma A_i = 1, 2, 3, 0, \dots$, a non-propagation situation with a single decoding error occurring when $\Sigma A_i = 3$. Note that this behavior is precisely that of the correctable burst $11 = i_j c_j$ considered at the beginning of this section.

Experimental work which bears out the results of this paper (and, in fact, motivated the research reported here) has already been published by the author [7].

V. CONCLUSION

A scheme has been developed for detecting a large majority of threshold decoder errors within a few digits after they have occurred. An added benefit is the prevention of error propagation. Although the example used to illustrate the discussion was for a rate one-half trial-and-error code, the method clearly applies to other classes of codes which are useful for threshold decoding [2].

REFERENCES

1. K. E. Perry and J. M. Wozencraft, "SECO: A self-regulating error-correcting coder-decoder," IRE Trans. Information Theory, vol. IT-8, pp. S128-S135, September 1962.
2. J. L. Massey, Threshold Decoding. Cambridge, Mass.: M.I.T. Press, 1963.
3. J. M. Wozencraft and B. Reiffen, Sequential Decoding, Cambridge, Mass.: M.I.T. Press, and New York: Wiley, 1961.
4. J. P. Robinson, "Error propagation and definite decoding of convolutional codes," IEEE Trans. Information Theory, vol. IT-14, pp. 121-128, January 1968.
5. D. D. Sullivan, "A fundamental inequality between the probabilities of binary subgroups and cosets," IEEE Trans. Information Theory, vol. IT-13, pp. 91-94, January 1967.
6. J. P. Robinson and A. J. Bernstein, "A class of binary recurrent codes with limited error propagation," IEEE Trans. Information Theory, vol. IT-13, pp. 106-113, January 1967.
7. D. Wiggert, "Error propagation and recovery in two decoding schemes using convolutional codes," Ph.D. dissertation, Dept. of Elec. Engrg., University of Wisconsin, Madison, September 1965.

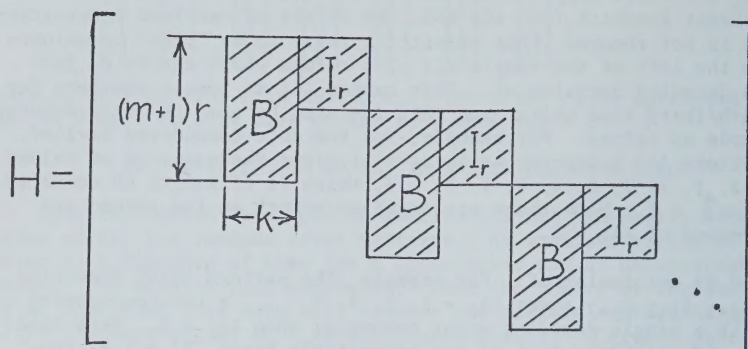
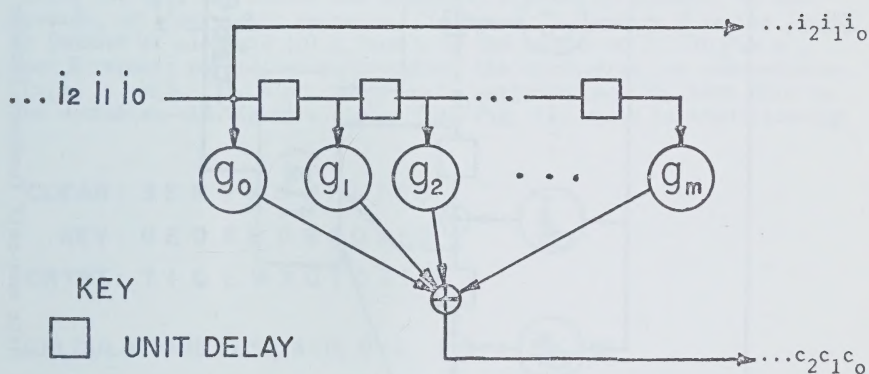


Figure 1. Structure of Parity-Check Matrix

$$\begin{bmatrix}
 g_o^{(1)} & g_o^{(2)} & \dots & g_o^{(k)} \\
 h_o^{(1)} & h_o^{(2)} & \dots & h_o^{(k)} \\
 \cdot & \cdot & \dots & \cdot \\
 z_o^{(1)} & z_o^{(2)} & \dots & z_o^{(k)} \\
 g_1^{(1)} & g_1^{(2)} & \dots & g_1^{(k)} \\
 \cdot & \cdot & \dots & \cdot \\
 z_1^{(1)} & z_1^{(2)} & \dots & z_1^{(k)} \\
 \cdot & \cdot & \dots & \cdot \\
 g_m^{(1)} & g_m^{(2)} & \dots & g_m^{(k)} \\
 \cdot & \cdot & \dots & \cdot \\
 z_m^{(1)} & z_m^{(2)} & \dots & z_m^{(k)}
 \end{bmatrix}$$

Figure 2. Details of B matrix



KEY



UNIT DELAY



MULTIPLIER



MODULO 2 ADDER

Figure 3. Rate one-half encoder

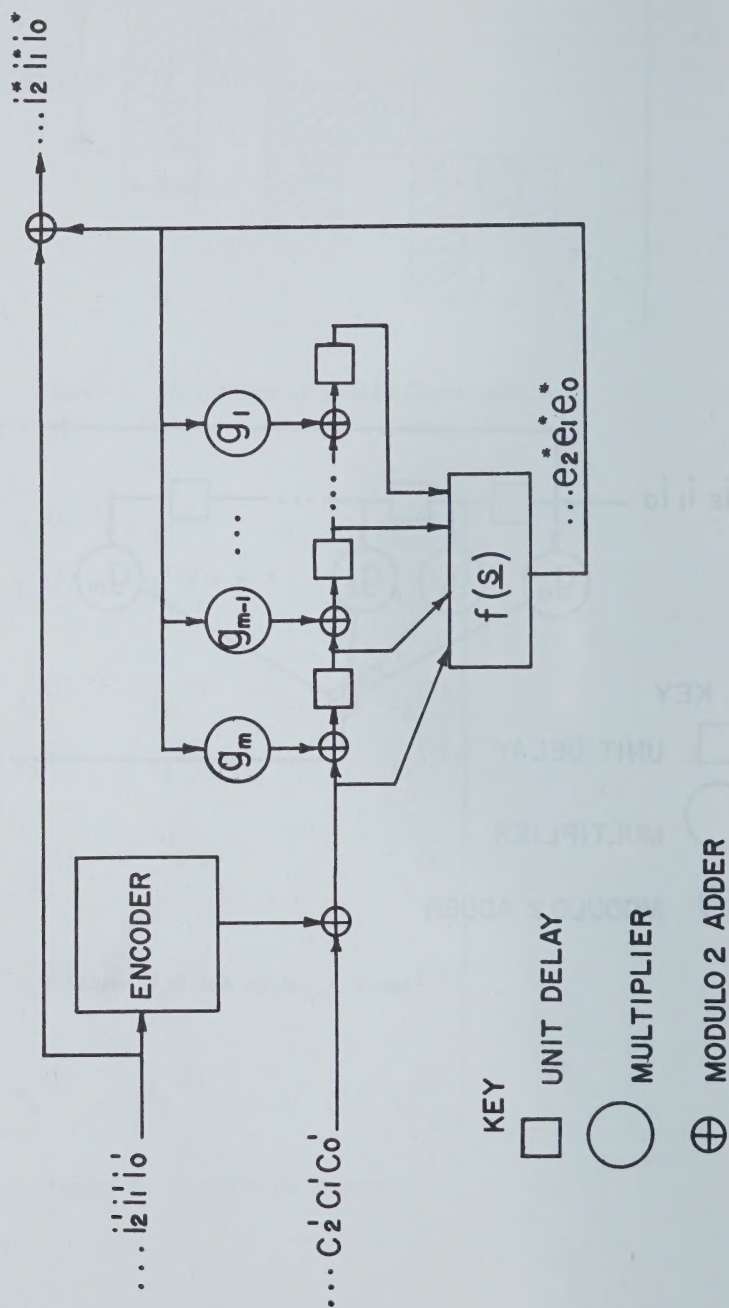


Figure 4. Rate one-half feedback decoder

SECURE ELECTRONIC CRYPTOGRAPHY

PHILIP R. GEFTE
Westinghouse Electric Corporation
Baltimore, Maryland

ABSTRACT

Digital electronics enables design of cryptographic machines which far surpass the capabilities of the electromechanical devices now in use. A major technical advance here comes from the theory of pseudo-random shift register sequences. The ability of digital machines to implement complicated systems reliably will make cryptanalysis a practical impossibility.

1. INTRODUCTION

A basic cryptographic method consists of generating a key sequence, $K(k_1 k_2 \dots k_\ell)$, and adding the sequence, element by element, to the message, or clear text, sequence, $C(c_1 c_2 \dots c_t)$, where ℓ is the length of (number of elements in) K , and t is the length of C . If $\ell \ll t$, then K repeats periodically throughout the cryptogram, or sum-sequence, $X(x_1 x_2 \dots x_t)$. If, also, addition is performed mod 26, then this is the centuries-old Vigenère cipher (see Fig. 1), which is traditionally

CLEAR: S E N D S U P P L I E S

KEY: B E D B E D B E D B E D

CRYPT: T I Q E W X Q T O J I V

CALCULATIONS (USE A=0, B=1, . . . , Z = 25)

18	04	13	03	18	20	15	15	11	08	04	18
01	04	03	01	04	03	01	04	03	01	04	03
19	08	16	04	22	23	16	19	14	09	08	21

FIG. 1 THE VIGENERE CIPHER

performed with the help of a tableau of letters. The tableau is actually nothing more than a mod 26 addition table with the mod 26 numbers represented by the letters of the alphabet.

It is usual for the cryptographer first to assign an arbitrary one-to-one correspondence between the letters of the alphabet and the mod 26 numbers. It is therefore more accurate to represent the cryptogram as a compound cipher formed in two steps. The first step transforms C into E by the transformation T_1 (simple substitution):

$$T_1: C(c_1 \dots c_t) \longrightarrow E(e_1 \dots e_t) \quad (1)$$

In the second step, $E \rightarrow X$ under T_2 :

$$T_2: x_i = e_i + k_i \quad (2)$$

with modular addition being understood. Writing the cryptogram as a product of transformations gives:

$$X = T_2 \cdot T_1 \cdot C \quad (3)$$

Vigenère ciphers are easily broken [1] without knowledge of either of the transformations T_1, T_2 . But if K is non-periodic ($\ell \geq t$) and, especially, if K is incoherent, then cryptanalysis becomes difficult. By "incoherent," we mean simply that a cursory examination of K shows no regularity.

If K is generated by a random process (e.g., by a roulette wheel) and used only once, then the cipher is known to be unbreakable. [2] This unbreakable system is, nevertheless, not very useful because heavy traffic consumes an inordinate amount of key material. Accordingly, a popular method is to generate K by an explicit set of rules rather than by a random process. Incoherence is obtained by selecting the rules in such a way that the statistical properties of K are similar to those of a random sequence, i.e., K is then a pseudo-random sequence. [3] Key generation may then be thought of as a "randomizing" function, F_r :

$$K = F_r(K_m) \quad (4)$$

where K_m , the message key, is a short key sequence which determines the long key sequence, K . In practice, K_m may be nothing more than the initial state of a linear sequential machine, such as a linear feedback shift register (LFSR), or the electromechanical machine described below.

2. AN INSECURE CIPHER

Let E be a binary sequence of standard 5-bit code words obtained from using an ordinary teletypewriter, and let K be obtained from an LFSR with n cells, or stages. If the shift register is connected as a maximal-length generator, then

$$\ell = 2^n - 1 \quad (5)$$

and very long sequences are easily obtained. [3], [4]

Cryptanalysis of this system is not difficult. The analyst starts with a probable word which he knows must occur somewhere in the message. [1] Since T_1 is the standard teletype code, he can confidently substitute 5-bit binary groups for the known piece of clear-text, and write

$$k_i = x_i - e_i \quad (6)$$

If the k_i give him $2n$ successive bits of K , then the LFSR connections can be reconstructed, and the entire message read as soon as the probable word has been correctly positioned.

3. BETTER CIPHERS

Electromechanical Machines [5] A commercially available device consists of a set of keywheels, each keywheel having θ_i possible angular positions. The internal state of the machine at any given moment is specified by the sequence of angular displacements of the wheels. The machine is so arranged that it produces a periodic linear output sequence having a very long period. Specifically, if the θ_i are all relatively prime, then the period is the product of all these numbers. (One commercial product has a period of $47 \cdot 43 \cdot 41 \cdot 37 \cdot 31 \cdot 29 = 2.75 \times 10^9$.)

These machines can be modeled electronically as shown in Fig. 2.

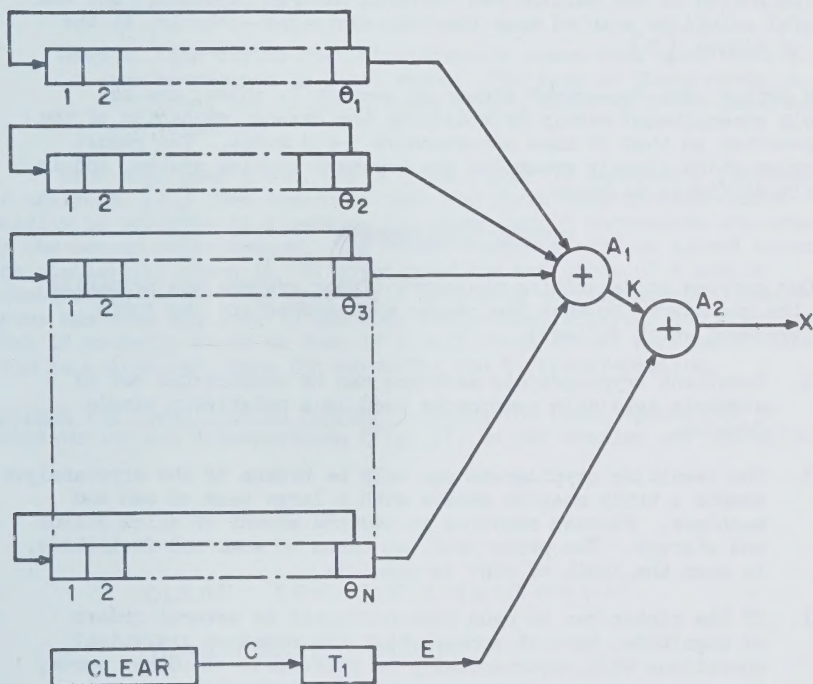


FIG. 2 ELECTRONIC MODEL OF KEYWHEEL MACHINE

Here, each of the N circular shift registers stores mod 26 numbers. The outputs of all the registers are added mod 26 to produce the key stream K , which is added in turn to the message stream E in the second adder. The output stream is the crypto-sequence, X .

The cryptographic variables of such a machine consist of the ordering of the numbers in each register, the initial set of phase angles, and the simple-substitution, T_1 .

Despite the impenetrable appearance of this formidable system, it can be broken by systematic analysis under heavy traffic conditions.[6] The details of the analytical method are not available, but the general outline is known. First, sets of equations relating the unknown key-variables are written based on the probable-word method. At the beginning, the cryptanalyst does not know either how many variables are independent, or how many of his equations are correct. When a sufficient amount of intercepted material has accumulated, various sets of equations are solved simultaneously. Success in this trial-and-error effort is indicated when some of the equation sets are seen to be consistent and, finally, when portions of coherent clear text begin to emerge from the intercepted cryptograms.

A monumental labor of this type is known to have solved Japanese keywheel-machine ciphers before and during World War II. According to Kahn, the period of the machine was "several hundred thousand," and the successful solutions enabled many American victories—notably, at the Battle of Midway. [7]

An ISFR Method The "insecure" cipher in section 2, above, can be immensely strengthened merely by modifying the key-bar mechanism of the teletypewriter so that it uses non-standard 5-bit words. The result is a system which closely resembles the keywheel machine system, and is similarly difficult to break.

4. FURTHER IMPROVEMENTS

Our purpose in describing the above cipher systems was primarily to lay the groundwork so that the reader will appreciate the force of the argument which follows:

- I. Excellent cryptographic machines can be constructed out of commonly available components used in a relatively simple fashion.
- II. The resulting cryptograms can only be broken if the cryptanalyst mounts a truly massive attack with a large team of men and machines. Success requires an immense amount of calculations and storage. The price paid, in terms of work and facilities, is near the limit of what is possible.
- III. If the cipher can be made more difficult by several orders of magnitude, then it appears that the required analytical operations will never actually be performed. In other words, the cipher will be "unbreakable" because it will never, in fact, be broken. There are many resources for accomplishing this result, including those described below.

Improvements by Conventional Cryptography If a cipher consists of two successive transformations on the clear text, then it is only common sense to require that both should, individually, be as difficult to analyze as possible. While it is obvious, this principle has not been widely applied because of the inherent limitations of mechanical machines, which have dominated the heavy-traffic applications for about forty years.

But it is obvious, to any engineer who thinks about it, that electronic digital equipment can perform any cryptographic operation that the cryptographer can imagine. The weakest link in the above systems is the simple substitution, T_1 , which, when used alone, can be solved by some amateurs almost as fast as they can write the solution down on paper. Better systems are widely known.

The deVries Cipher [8] This is a well known cipher which consists of two operations:

Step 1. Binary digits of varying group length are substituted for the letters of the clear text. The groups have the comma-free property. This means that the spaces between groups can be restored by inspection after transmission without spacing. Many distinct substitutions of this type are possible. [9]

Step 2. The digits resulting from the comma-free substitution are re-grouped in 5-bit words. For each of these words, a different one is substituted from a table of 5-bit-for-5-bit equivalences.

Even when used alone, the deVries cipher is remarkably resistant to analysis. [10] The reason is that the first substitution cannot readily be parenthesized because the group-length indicators are erased by the second substitution. The latter cannot easily be solved because the statistical clues in the cryptogram are not those of a simple substitution cipher. No systematic method of solving deVries cryptograms has ever appeared in the open literature, although one presumes that it probably could be done if a sufficient effort were made. Clearly, this is a good candidate for upgrading the T_1 transformation.

Through the "Complication Barrier" Using the basic operations of substitution and transposition (Fig. 3), we can overlay one operation

CLEAR: "SEND SUPPLIES TO P . . ."

TRANSPOSITION BLOCK:

S E N D S
U P P L I
E S T O P

TRANSPPOSED TEXT:

P I S O L D T P N S P E E U S

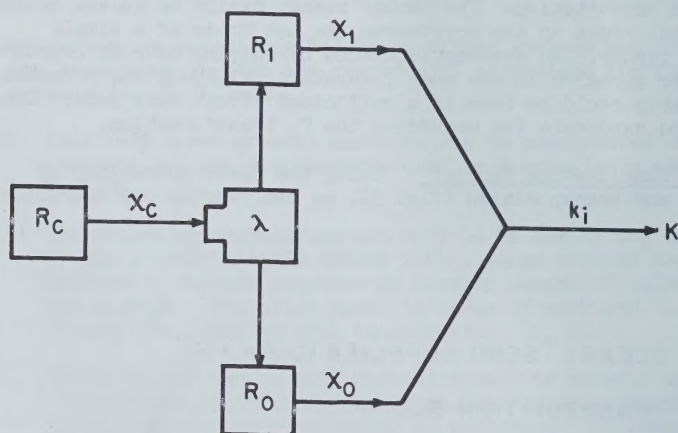
FIG. 3 EXAMPLE OF TRANSPOSITION

onto another so as to greatly increase the security of the cipher. This has not been possible with mechanical machines, and it has not been practical with paper-and-pencil ciphers because the inevitable errors result in a garbled cryptogram.

But digital equipment can perform complicated operations without error, and the basic operations of cryptography are easy to implement. For example, transposition is accomplished by a rectangular matrix (array of shift registers) wired for parallel read-in and serial read-out. Also, provision can be made for dynamic programming of transposition by pseudo-random sequences.

Non-Linear FSR Sequences The difficulty of unravelling a long key sequence can be enormously increased by using non-linear sequence generators. In general, these sequences do not have the regular properties of the linear sequences. They do not usually have flat distributions or the shift-and-add property. They can be generated in an infinite number of ways, and the generator cannot be reconstructed from one of its sequences merely by solving linear equations.

Non-linear generators are shown in Figures 4 and 5. If non-linear

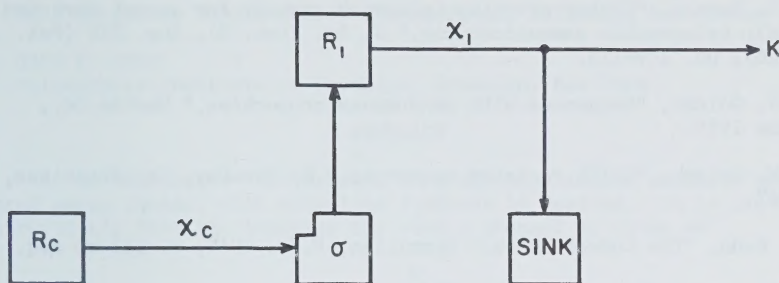


R_C, R_1, R_0 ARE LINEAR GENERATORS

IF $x_C = 1$, THEN $k_i = x_1$

IF $x_C = 0$, THEN $k_i = x_0$

FIG. 4. A NON-LINEAR GENERATOR



R_1, R_c ARE LINEAR GENERATORS
 IF $x_c = 1$, THEN x_1 GOES INTO K
 IF $x_c = 0$, THEN x_1 GOES INTO
 THE BIT SINK (IS DELETED)

FIG. 5 ANOTHER NON-LINEAR GENERATOR

operations are applied to linear sequences, it is easy to assure long periods in the non-linear result, and this scheme is employed in the figures. These suggested methods are illustrative only, and an unlimited variety is easily obtainable by using a little imagination.

5. CONCLUSIONS

Digital electronics has made it possible to break the monopoly previously enjoyed by electromechanical cryptographic machines. Because electronic machines are error-free and more versatile, we can take advantage of many well known artifices which will increase message security beyond the point where cryptanalysis is feasible. A part of the improvement comes from conventional cryptography, and a part from the last ten years of progress in the theory of linear and non-linear shift-register sequences.

While the best imaginable cryptographic machine would be a large, general-purpose digital computer, small special-purpose machines can implement very complicated ciphers inexpensively and reliably.

6. REFERENCES

1. H.F. Gaines, "Cryptanalysis," Dover, N.Y., 1956.
2. G.S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," J. Am. Inst. El. Eng. XLV (Feb. 1926), pp. 109-115.
3. S.W. Golomb, "Sequences with randomness properties," Martin Co., June 1955.
4. S.W. Golomb, "Shift register sequences," Holden-Day, San Francisco, 1967.
5. D. Kahn, "The Codebreakers," Macmillan, N.Y., 1967, p. 411 et seq.
6. *ibid*, pp. 1-67.
7. *ibid*, pp. 556-573.
8. M. deVries, "Kwantificering in taal en cryptoanalyses," Statistica Neerlandica, vol. 7, no. 4 (1953), pp. 233-256.
9. C.V. Ramamoorthy and D.W. Tufts, "Reinforced prefixed comma-free codes," IEEE Trans. Inf. Theory, July 1967, pp. 366-371.
10. Homo Sapiens, "The deVries cipher," The Cryptogram, vol. XXVIII, no. 1 (Sept-Oct. 1960), pp. 10-12 (pub. by the Am. Cryptogram Assoc.).

CAPACITY OF AN ADDITIVE GAUSSIAN COLORED NOISE CHANNEL WITH NOISELESS FEEDBACK*

THOMAS W. EDDY

Bell Telephone Laboratories, Incorporated, Whippany, New Jersey

JACK K. WOLF

Polytechnic Institute of Brooklyn, Brooklyn, New York

ABSTRACT

In this paper the capacity of a non-bandlimited additive gaussian colored noise channel with noiseless feedback is derived. It is proved constructively that the capacity for such a channel is given as

$$C_{fb} = \frac{P_{avg}}{2 \min_{\omega} S_n(\omega)}$$

where P_{avg} is the average transmitted power and $S_n(\omega)$ is the spectral density of the additive noise in the forward channel. With this result it follows that the capacity of the channel with feedback can be larger than the capacity of the corresponding channel without feedback.

1. INTRODUCTION

In this paper, in a relatively simple manner, the capacity of the additive gaussian colored noise channel with no bandwidth constraint is calculated for the case where a noiseless feedback link is available from the receiver to transmitter. The procedure used is as follows:

- a) A simple modification of the Schalkwijk-Kailath scheme [1] (which was originally applied to the additive white noise channel) is analyzed for the case of additive colored noise. A maximum rate, R_{max}^* , is determined such that for rates less than R_{max}^* , the error probability goes to zero as the coding block length, N , goes to infinity.
- b) It is then proved that the capacity for this channel with feedback is upper bounded by R_{max}^* .

Since a constructive scheme is presented for transmitting at rates less than R_{max}^* with vanishingly small error probability and since the capacity of the channel with feedback is at most R_{max}^* then we have established R_{max}^* as the capacity of this channel with feedback.

* Taken from the dissertation submitted to the Faculty of the Polytechnic Institute of Brooklyn in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Electrical Engineering), June 1968.

This research was partially supported by the Air Force Office of Scientific Research under Contract AF 49(638)-1600.

No claim for optimality is made for the coding scheme utilized. In fact, the scheme seems ill-fitted to a colored noise channel since no attempt is made to optimize the design of the transmitted waveforms (except in a very simple manner). On the other hand, the above results show that this simple modification does "achieve capacity" for this channel, although other schemes may have a smaller error probability for equal block lengths.

2. DESCRIPTION OF CODING SCHEME

In reading this section it will be helpful to consult the block diagram shown in Figure 1. The transmitter is allowed T seconds to send one of M equally likely signals $\theta_i = \frac{1}{M} - \frac{1}{2M}$, $i = 1, 2, \dots, M$. The trans-

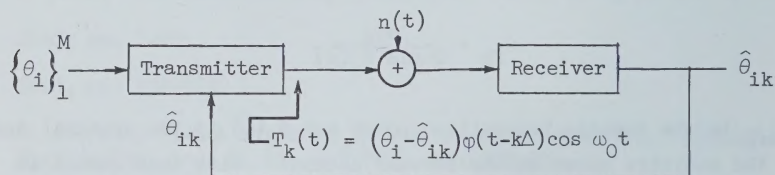


FIGURE 1

The Communication System for $k\Delta \leq t < (k+1)\Delta$

mission of this one signal is accomplished in N steps, each step taking Δ seconds. Thus $T = N\Delta$. Transmission begins at time $t = 0$ and for the first Δ seconds, the transmitter sends the signal.

$$T_0(t) = (\theta_i - 0.5)\varphi(t)\cos \omega_0 t$$

where

$$\varphi(t) = \begin{cases} \frac{1}{\sqrt{\Delta}} & 0 \leq t < \Delta \\ 0 & \text{elsewhere} \end{cases}$$

and ω_0 is an arbitrary constant, for which a good choice will be made later. After the first Δ seconds, the receiver computes an estimate of θ_i (which we call $\hat{\theta}_{i1}$) and relays this information back to the transmitter via a noiseless feedback channel. The transmitter, then modifies its output for the next Δ seconds to be

$$T_1(t) = (\theta_i - \hat{\theta}_{i1})\varphi(t - \Delta)\cos \omega_0 t.$$

This process is repeated, whereby after every transmission, the receiver computes a new estimate of θ_i , which is then used by the transmitter in the next transmission. We denote as $\hat{\theta}_{ik}$, the receiver's estimate after $k\Delta$

seconds of transmission ($\hat{\theta}_{i0} = 0.5$) and in the time interval $(k\Delta, (k+1)\Delta)$ the transmitter's output is

$$T_k(t) = (\theta_i - \hat{\theta}_{ik}) \varphi(t - k\Delta) \cos \omega_0 t, \quad k = 0, 1, \dots, N-1.$$

The only effect of the channel on the transmitted waveform is to add noise $n(t)$ with zero mean and power spectral density $S_n(\omega)$. Thus the received waveform, $r(t)$ is given as

$$r(t) = \sum_{k=0}^{N-1} T_k(t) + n(t).$$

The receiver's estimate after $q\Delta$ seconds is taken to be

$$\hat{\theta}_{iq} = \frac{1}{q} \sum_{k=0}^{q-1} \left[2 \int_{k\Delta}^{(k+1)\Delta} r(t) \varphi(t - k\Delta) \cos \omega_0 t \, dt + \frac{2}{\Delta} \int_{k\Delta}^{(k+1)\Delta} \hat{\theta}_{ik} \cos^2 \omega_0 t \, dt \right].$$

After $N\Delta$ seconds, the receiver chooses as the transmitted message, the signal θ_j having the smallest absolute difference from the estimate $\hat{\theta}_{iN}$. The probability of error for this scheme is then

$$P_E = \text{Prob}[\theta_j \neq \theta_i].$$

3. PROBABILITY OF ERROR

To evaluate the probability of error for the coding scheme, we proceed to find the mean and variance of $\hat{\theta}_{iq}$, the estimator of θ_i after $q\Delta$ seconds of transmission.

Substituting for $r(t)$ in the expression for $\hat{\theta}_{iq}$ gives

$$\hat{\theta}_{iq} = \frac{2\theta_i}{q\Delta} \int_0^{q\Delta} \cos^2 \omega_0 t \, dt + \frac{2}{q\sqrt{\Delta}} \int_0^{q\Delta} n(t) \cos \omega_0 t \, dt. \quad (3.1)$$

Taking the expectation of $\hat{\theta}_{iq}$ and performing the integration, it follows that

$$E\{\hat{\theta}_{iq}\} = \theta_i \left[1 + \frac{\sin 2\omega_0 q\Delta}{2\omega_0 q\Delta} \right].$$

Clearly

$$\lim_{q\Delta \rightarrow \infty} E\{\hat{\theta}_{iq}\} = \theta_i.$$

In the Appendix, the asymptotic variance of $\hat{\theta}_{iq}$ (large q) is shown to be

$$\text{var}\{\hat{\theta}_{iq}\} \sim \frac{2S_n(\omega_0)}{q}.$$

Thus, the probability of error is

$$P_E \sim 2Q\left(\frac{1}{2M} \sqrt{\frac{N}{2S_n(\omega_0)}}\right)$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-y^2/2} dy.$$

The probability of error will go to zero as $N \rightarrow \infty$ provided

$$M < N^{1/2}$$

or, using the definition of rate, when

$$R = \frac{\ln M}{T} < R^* = \frac{1}{2} \frac{\ln N}{N\Delta}.$$

For R^* to remain constant as $N \rightarrow \infty$, N must satisfy

$$N = e^{2R^*T}.$$

Note that since

$$T = N\Delta = \Delta e^{2R^*T}$$

from which

$$\Delta = T e^{-2R^*T}$$

we see that although $N \rightarrow \infty$ as $T \rightarrow \infty$, $\Delta \rightarrow 0$ as $T \rightarrow \infty$.

4. THE MAXIMUM RATE $R_{\max}^*$

To express R^* in the usual form we evaluate the average transmitted power, P_{avg} , given by

$$P_{\text{avg}} = \frac{1}{T} \int_0^T E\{T^2(t)\} dt$$

where

$$T(t) = \sum_{k=0}^{N-1} T_k(t).$$

Substituting for $T(t)$, this becomes (for large T)

$$P_{\text{avg}} = \frac{1}{2T} \sum_{k=0}^{N-1} \text{var}\{\hat{\theta}_{ik}\} (1 + \cos 2\omega_0 k\Delta)$$

Now using the asymptotic results for $\text{var}\{\hat{\theta}_{ik}\}$

$$P_{\text{avg}} \sim S_n(\omega_0) \frac{\ln N}{T} = 2S_n(\omega_0)R^*$$

from which

$$R^* = \frac{P_{\text{avg}}}{2S_n(\omega_0)}.$$

Since ω_0 is arbitrary, we may maximize R^* by choosing ω_0 to coincide with the minimum value of the noise spectral density. Thus[†]

$$R_{\text{max}}^* = \frac{P_{\text{avg}}}{2 \min_{\omega} S_n(\omega)} \quad (4.1)$$

5. UPPER BOUND ON CHANNEL CAPACITY WITH FEEDBACK

The additive gaussian colored noise can be considered as the sum of two separate uncorrelated gaussian noise sources $n_1(t)$ and $n_2(t)$ with power spectral densities

[†] Although equation (4.1) is valid for $\omega_0 = 0$ or $\omega_0 = \infty$, the coding scheme would require some minor modifications. It is shown in [2] that the Schalkwijk-Kailath coding scheme [1] using particular waveforms $\varphi(t)$, can transmit over the colored noise channel with arbitrary probability of error at rates less than

$$R_{\text{max}}^* = \frac{P_{\text{avg}}}{2S_n(0)}$$

or

$$R_{\text{max}}^* = \frac{P_{\text{avg}}}{2S_n(\infty)}.$$

$$S_{n_1}(\omega) = \min_{\omega} S_n(\omega) \equiv \frac{N_0}{2} \quad \text{for all } \omega$$

and

$$S_{n_2}(\omega) = S_n(\omega) - \min_{\omega} S_n(\omega) = S_n(\omega) - \frac{N_0}{2}.$$

Consider a system with additive white noise $n_1(t)$ in the forward channel. The capacity of this hypothetical channel without feedback is given by [3]

$$C = \frac{P_{\text{avg}}}{N_0}$$

and furthermore, the capacity cannot be increased by feedback regardless of the signal processor at the receiver [4]. Therefore the channel capacity cannot be increased by a processor which adds the noise $n_2(t)$ to its input and then uses noiseless feedback. It follows that the capacity of the colored noise channel with feedback cannot be greater than P_{avg}/N_0 : i.e.,

$$C_{\text{fb}} \leq \frac{P_{\text{avg}}}{N_0} = \frac{P_{\text{avg}}}{2 \min_{\omega} S_n(\omega)}.$$

6. CAPACITY OF THE COLORED NOISE CHANNEL WITH FEEDBACK

In section 2 we have given a constructive technique employing noiseless feedback for achieving an arbitrarily small error probability at rates such that

$$R < R_{\text{max}}^* = \frac{P_{\text{avg}}}{2 \min_{\omega} S_n(\omega)}.$$

Therefore the capacity of this channel with feedback is at least R_{max}^* . In section 5 we have shown that the capacity of the channel with noiseless feedback is bounded as

$$C_{\text{fb}} \leq \frac{P_{\text{avg}}}{2 \min_{\omega} S_n(\omega)}.$$

Thus we have the following conclusion which we state as a theorem.

Theorem: Given an additive noise channel with power spectral density $S_n(\omega)$, and an average transmitted power P_{avg} , the capacity of this channel with noiseless feedback is given as

$$C_{fb} = \frac{P_{avg}}{2 \min_{\omega} S_n(\omega)}.$$

Furthermore, a deterministic scheme is available for achieving transmissions at arbitrarily small error probabilities at rates below this capacity.

The capacity of a gaussian, colored noise channel without feedback is given as [3]

$$C = \frac{1}{4\pi} \int_{-\infty}^{\infty} \ln \left[1 + \frac{S_s(\omega)}{S_n(\omega)} \right] d\omega$$

where $S_s(\omega)$ is the spectral density of the transmitted signal and

$$S_s(\omega) = \begin{cases} B - S_n(\omega); & S_n(\omega) < B \\ 0 & ; \quad S_n(\omega) \geq B \end{cases}$$

where B is a constant such that

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} S_s(\omega) d\omega = P_{avg}.$$

It is easy to construct examples of colored noise where the capacity of the channel without feedback is strictly less than the capacity of the channel with feedback. Thus, it is possible to increase the capacity of some channels by adding a noiseless feedback channel and, at least for the case of additive gaussian colored noise, a quantitative measure of this increase is available.

APPENDIX

In this appendix we shall obtain the asymptotic behavior of $\text{var}\{\hat{\theta}_{iq}\}$ for large values of q .

Using the expression for $\hat{\theta}_{ik}$ given by (3.1), it follows directly that

$$\text{var}\{\hat{\theta}_{iq}\} = \frac{4}{q^2 \Delta} \int_0^{q\Delta} \int_0^{q\Delta} R_n(t_1 - t_2) \cos \omega_0 t_1 \cos \omega_0 t_2 dt_1 dt_2.$$

In terms of the noise spectral density $S_n(\omega)$,

$$\text{var}\{\hat{\theta}_{iq}\} = \frac{4}{q^2\Delta} \int_0^{q\Delta} \int_0^{q\Delta} \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} S_n(\omega) e^{j\omega(t_1-t_2)} d\omega \right] \cos \omega_0 t_1 \cos \omega_0 t_2 dt_1 dt_2.$$

Interchanging the order of integration and recalling $t = q\Delta$,

$$\text{var}\{\hat{\theta}_{iq}\} = \frac{2}{q} \int_{-\infty}^{\infty} S_n(\omega) \left[\frac{1}{\pi t} \left| \int_0^t \cos \omega_0 y e^{-j\omega y} dy \right|^2 \right] d\omega. \quad (\text{A.1})$$

We digress to investigate the bracketed term in the last equation. After integration it can be expressed as

$$\begin{aligned} \left[\frac{1}{\pi t} \left| \int_0^t \cos \omega_0 y e^{-j\omega y} dy \right|^2 \right] &= \frac{1}{\pi t} \left[\frac{\sin^2(\omega - \omega_0)t/2}{(\omega - \omega_0)^2} + \frac{\sin^2(\omega + \omega_0)t/2}{(\omega + \omega_0)^2} \right] \\ &\quad + \frac{1}{\pi t} \frac{2 \sin(\omega - \omega_0)t/2 \sin(\omega + \omega_0)t/2}{(\omega - \omega_0)(\omega + \omega_0)} \cos \omega_0 t \end{aligned}$$

Treating this as a distribution, we have

$$\lim_{t \rightarrow \infty} \left[\frac{1}{\pi t} \left| \int_0^t \cos \omega_0 y e^{-j\omega y} dy \right|^2 \right] = \frac{\delta(\omega - \omega_0)}{2} + \frac{\delta(\omega + \omega_0)}{2}.$$

Using this last result in equation (A.1) yields

$$\lim_{t \rightarrow \infty} q \text{var}\{\hat{\theta}_{iq}\} = 2S_n(\omega_0)$$

from which we obtain the desired result that

$$\text{var}\{\hat{\theta}_{iq}\} \sim \frac{2S_n(\omega_0)}{q}.$$

REFERENCES

- [1] J. P. M. Schalkwijk and T. Kailath, "A Coding Scheme for Additive Noise Channels with Feedback - Part I: No Bandwidth Constraint", IEEE Transactions on Information Theory, Vol. IT-12, No. 2 (April 1966) pp. 172-182.
- [2] T. W. Eddy, "Some Results for Additive Noise Channels with Noiseless Information Feedback", Polytechnic Institute of Brooklyn Ph.D. Thesis (Electrical Engineering) June 1968.

- [3] R. M. Fano, The Transmission of Information, M.I.T. Press, Cambridge, Mass., 1961.
- [4] C. E. Shannon, "The Zero-Error Capacity of a Noisy Channel", IRE Transactions on Information Theory, vol. IT-12, No. 3, pp. 8-19; September 1956.

ON THE CONSISTENCY CONDITIONS AND THE CANONICAL FORM OF FIXED LINEAR DYNAMIC SYSTEMS*

ANTHONY J. GRAMMATICOS
New York University
Department of Electrical Engineering
Bronx, New York

ABSTRACT

A given physical fixed linear dynamic system is mathematically described by a system of ordinary differential equations. Here the consistency conditions and the canonical form for the system are discussed using the theory of λ -matrices and elementary divisors.

I. INTRODUCTION

For a given physical linear dynamic system, one obtains a system of scalar ordinary differential equations which mathematically describe the behavior of the system by applying the appropriate physical laws. In general, the resulting system of differential equations is of the form $F\dot{\underline{x}} + G\underline{x} = H\underline{v}(I)$, where $\underline{v}$ is the input, $\underline{x}$ is a vector of dependent variables, and F, G, H are matrices of appropriate sizes.

Now in modern system theory, linear dynamic systems are described mathematically by the state differential equations $\dot{\underline{q}} = A\underline{q} + B\underline{u}$ (II), where $\underline{u}$ is the input and $\underline{q}$ is the state vector^[1,2].** Obviously, equation (II) can be obtained from equation (I) by premultiplying the latter by F^{-1} , provided that F is a square nonsingular matrix. However, this is not true in general. The matrix F may often be either nonsquare or square and singular. In such a case the transformation of the system (I) into the system (II) is not trivial.

In this paper, the system (I) is assumed to consist of m equations in n dependent variables (i.e., $\underline{x}$ is an n -vector) and F, G, H are arbitrary nonsquare matrices of appropriate size. Such assumptions are very general and realistic as, for example, in the case where one has to choose states and write the differential equations describing a given network.

Using the theory of λ -matrices and elementary divisors^[3], it is shown that the system (I) can be decomposed into several parts, one of which is in the form of system (II) containing the dynamics of the original system. The remaining parts consist (in general) of chains of open loop integrators and differentiators. It is also shown that the components of the input $\underline{v}$ cannot be completely arbitrary but they must satisfy certain consistency conditions for a unique solution to exist. An illustrative example is given.

II. THE CANONICAL FORM AND THE CONSISTENCY CONDITIONS

Assume the behavior of a physical system to be described mathematically by the system of linear ordinary differential equations.

$$F\dot{\underline{x}}(t) + G\underline{x}(t) = H\underline{v}(t) \quad (1)$$

where the dot represents differentiation with respect to time, $\underline{v}(t)$ is the input r -vector, $\underline{x}(t)$ is an n -vector of dependent variables and F, G, H are con-

*This research was supported in part by the U.S. Air Force, Air Force Office of Scientific Research under Grant AF-AFOSR 68-1393.

**Bracketed numbers indicate references at the end of this paper.

stant matrices of sizes $m \times n$, $m \times n$, $m \times r$ respectively.

Introducing the operator notation $p = \frac{d}{dt}$ Eq. (1) becomes

$$\begin{aligned} Fp\underline{x} + G\underline{x} &= H\underline{v} \\ (Fp+G) \underline{x} &= H\underline{v}. \end{aligned} \quad (2)$$

Let P and Q be two square constant nonsingular matrices of order m and n respectively. In Eq. (2) introduce the change of variables

$$\underline{x} = Q\underline{y} \quad (3)$$

and premultiply both sides by P ; we obtain

$$P(Fp+G) Q\underline{y} = PH\underline{v}. \quad (4)$$

The linear binomial in p , $Fp+G$ can be considered as a pencil of matrices*. Furthermore by Definition A.4, it is strictly equivalent to

$$F_1p + G_1 = P(Fp+G)Q \quad (5)$$

since both P and Q are constant nonsingular matrices. For different choices of P and Q we obtain different pencils F_1p+G_1 all of them strictly equivalent to the pencil $Fp+G$. Here choose the particular P and Q which according to Theorem A.5, give the pencil F_1p+G_1 the canonical quasi-diagonal form

$$F_1p + G_1 = \{\theta, L_{\epsilon_{g+1}}, \dots, L_{\epsilon_k}, L_{\eta_{h+1}}^T, \dots, L_{\eta_\ell}^T, N_{t_1}, \dots, N_{t_s}, J+pI\} \quad (6)$$

which corresponds to the complete systems of minimal indices

$$\epsilon_1 = \dots = \epsilon_g = 0, \quad 0 < \epsilon_{g+1} \leq \dots \leq \epsilon_k \quad (7)$$

$$\eta_1 = \dots = \eta_h = 0, \quad 0 < \eta_{h+1} \leq \dots \leq \eta_\ell \quad (8)$$

and the infinite elementary divisors $\mu^{t_1}, \mu^{t_2}, \dots, \mu^{t_s}$ of the pencil $Fp + G$; J is in Jordan normal form and I is the identity matrix.

In view of (5) and (6), Eq. (4) becomes

$$\{\theta, L_{\epsilon_{g+1}}, \dots, L_{\epsilon_k}, L_{\eta_{h+1}}^T, \dots, L_{\eta_\ell}^T, N_{t_1}, \dots, N_{t_s}, J+pI\}\underline{y} = \underline{u} \quad (9)$$

where we set

$$PH\underline{v} \triangleq \underline{u} \quad (10)$$

Comment: Since P and Q are constant nonsingular matrices, Eq. (5) expresses a change of coordinates. Now, Eq. (9) was obtained from Eq. (1) through such a change of coordinates; hence they both describe the same dynamic system in different coordinates.

Due to the quasi-diagonal form of (6), the system (9) is decomposed into several subsystems which can be integrated separately as follows:

Corresponding to $\epsilon_1 = \dots = \epsilon_g = 0$, $\eta_1 = \dots = \eta_h = 0$ we obtain

*For a definition of this and other related terms see the Appendix to this paper. For more details consult Reference 3.

$$\theta \underline{y}_0 = \underline{u}_0 \quad (11)$$

where

$\underline{y}_0$ is a g -vector, and
 $\underline{u}_0$ is an h -vector.

Eq. (11) is consistent if $\underline{u}_0 = \underline{0}$ and $\underline{y}_0$ is arbitrary.

Corresponding to $\varepsilon_{g+1}, \dots, \varepsilon_k$ we obtain $k - g$ decoupled systems of the form*

$$\dot{y}_1 + y_2 = u_1, \dot{y}_2 + y_3 = u_2, \dots, \dot{y}_\varepsilon + y_{\varepsilon+1} = u_\varepsilon \quad (12)$$

Except for an arbitrary choice of one of the unknown functions, say $y_{\varepsilon+1}$, such a system is always consistent and the remaining unknowns can be determined successively by direct integration. Figure 1a shows a simulation of system (12).

Corresponding to $\eta_{h+1}, \dots, \eta_\ell$ we obtain $\ell - h$ decoupled systems of the form

$$\dot{y}_1 = u_1, \dot{y}_2 + y_1 = u_2, \dots, \dot{y}_\eta + y_{\eta-1} = u_\eta, y_\eta = u_{\eta+1} \quad (13)$$

Successive elimination of the unknown functions from Eq. (13) leads to

$$u_1 - \dot{u}_2 + \dots + (-1)^\eta \frac{d^\eta}{dt^\eta} u_{\eta+1} = 0 \quad (14)$$

This is a consistency condition for Eq. (13); if it is satisfied, then Eq. (13) has a unique solution. An analogue computer simulation of Eq. (13) is shown in Fig. 1b.

Subsystems (11), (12), and (13) exist if the pencil $Fp + G$ has minimal indices. The pencil may also have infinite and finite elementary divisors; they give rise to subsystems of the following form.

Corresponding to the infinite elementary divisors $\mu^{t_1}, \mu^{t_2}, \dots, \mu^{t_s}$ we obtain s subsystems of the form

$$y_1 + \dot{y}_2 = u_1, y_2 + \dot{y}_3 = u_2, \dots, y_{t-1} + \dot{y}_t = u_{t-1}, y_t = u_t \quad (15)$$

Eqs. (15) can be solved successively and they yield a unique solution. Figure 1c shows a computer simulation of Eq. (15).

Finally, corresponding to the finite elementary divisors of $Fp + G$ we obtain the subsystem

$$\dot{\underline{y}} + J\underline{y} = \underline{u} \quad (16)$$

which is in the form used in the axiomatic definition of a dynamical system. Such a system has the unique solution

$$\underline{y}(t) = e^{-J(t-t_0)} \underline{y}(t_0) + \int_{t_0}^t e^{-J(t-\tau)} \underline{u}(\tau) d\tau, t \geq t_0 \quad (17)$$

*Here and in the following, the subscripts must be understood as running inside the individual matrix block under consideration; this simplifies the notation considerably.

Figure 1d is a block diagram representation of Eq. (16).

Table 1 summarizes most of the results obtained above.

TABLE 1

Subsystem Corresponds to	Form of subsystem	Open-loop or Closed-loop	Number of Arbitrary Functions $y(t)$	Number of Consistency Conditions on $u(t)$	Nature of Solutions
$\varepsilon_1 = \dots = \varepsilon_g = 0$ $\eta_1 = \dots = \eta_h = 0$	θ	open-loop	g	h	arbitrary
$\varepsilon_{g+1}, \dots, \varepsilon_k$	$L_\varepsilon(p)$	"	$k-g$	none	unique solution subject to the choice of arbitrary functions
$\eta_{h+1}, \dots, \eta_\ell$	$L_\eta^T(p)$	"	none	$\ell-h$	unique solution subject to the satisfaction of the consistency conditions
infinite E.D.	$N_t(p)$	"	none	none	unique
finite E.D.	$J + pI$	closed-loop	none	none	unique

III. AN EXAMPLE

As an example consider the system

$$F\dot{\underline{x}} + G\underline{x} = H\underline{v}$$

where

$$F = \begin{bmatrix} -2 & 8/3 & 23/28 & -1/12 \\ 1 & -4/3 & -15/28 & 1/12 \\ 2 & -8/3 & -3/7 & 0 \\ -2 & 0 & -7/28 & 1/12 \\ -1 & 4/3 & 11/28 & -1/12 \end{bmatrix}, \quad G = \begin{bmatrix} 9 & -31/3 & -6/28 & -1/6 \\ -6 & 7 & -2/28 & 1/6 \\ -5 & 17/3 & 4/7 & 0 \\ -2 & 7/3 & -6/28 & 1/6 \\ 4 & -14/3 & 2/28 & -1/6 \end{bmatrix}$$

$$H = \begin{bmatrix} 2 & -4 & 5 & -3 & -1/2 \\ -1 & 3 & -3 & 1 & 1/2 \\ -2 & 2 & -3 & 4 & 0 \\ 0 & 1 & -1 & 0 & 1/2 \\ 1 & -2 & 2 & -1 & -1/2 \end{bmatrix}, \quad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad \underline{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix}$$

For the particular choice of P and Q

$$P = \begin{bmatrix} 1 & 2 & 1 & 2 & 3 \\ 2 & 3 & 1 & 0 & 1 \\ 2 & 2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & -2 & 0 & 2 & -2 \end{bmatrix}, \quad Q = \begin{bmatrix} 3 & 4 & 0 & 0 \\ 3 & 3 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 9 & 6 \end{bmatrix}$$

$PH = I$ (i.e., $\underline{v} = \underline{u}$) and the pencil $F_1 p + G_1 = P (Fp+G)Q$ has the canonical quasi-diagonal form

$$P(Fp+G)Q = \begin{bmatrix} p: 0 & 0 & 0 \\ .1: .Q & .Q & 0 \\ 0: 1 & p: 0 \\ 0: .0 & 1: .0 \\ 0 & 0 & 0: p+2 \end{bmatrix}$$

This indicates that $Fp + G$ has only

- one minimal index $\eta = 1$
- one infinite elementary divisor μ^2 , and
- one finite elementary divisor $p + 2$.

The corresponding subsystems give

$$\begin{aligned} \dot{y}_1 &= u_1 \\ y_1 &= u_2 \end{aligned} \Rightarrow u_1 - \dot{u}_2 = 0; \text{ one consistency condition}$$

$$\begin{aligned} y_2 + \dot{y}_3 &= u_3 \\ y_3 &= u_4 \end{aligned} \Rightarrow \begin{aligned} y_2 &= u_3 - \dot{y}_3 = u_3 - \dot{u}_4 \\ y_3 &= u_4 \end{aligned} \} \text{ a unique solution}$$

$$\dot{y}_4 + 2y_4 = u_5 \Rightarrow y_4(t) = y_4(t_0)e^{-2(t-t_0)} + \int_{t_0}^t e^{-2(t-\tau)} u_5(\tau) d\tau, \quad t \geq t_0$$

a unique solution.

In other words the system is consistent if $u_1 - \dot{u}_2 = 0$ and it has one open-loop and one closed-loop solutions.

It is worth noting that the system has no minimal indices other than $\eta=1$.

IV. CONCLUSIONS

A physical system described mathematically by a system of fixed linear ordinary differential equations can be decomposed into several subsystems which can be integrated separately. The number and form of these subsystems is determined by the systems of minimal indices and the infinite and finite elementary divisors of the pencil of matrices obtained from the system of equations.

The components of the input vector must satisfy as many consistency conditions as the number of the minimal indices for the rows. On the other hand as many dependent variables as the number of minimal indices for the columns, can be chosen arbitrarily.

The infinite elementary divisors give rise to subsystems with unique open-loop solutions; the finite elementary divisors give rise to a sub-

system with unique closed-loop solution; it is only this last subsystem which corresponds to what is axiomatically defined as a Dynamical System.

REFERENCES

- [1] R.E. Kalman, Mathematical Description of Linear Dynamical Systems, J.S.I.A.M. Control, Series A, Vol. 1, No. 2, 1963.
- [2] L.A. Zadeh and C.A. Desoer, Linear System Theory, McGraw-Hill, New York, 1963.
- [3] F.R. Gantmakher, The Theory of Matrices, Vol. I and II, Chelsea, New York, 1959.

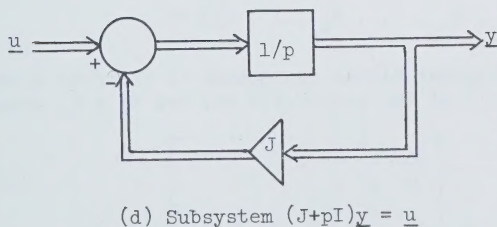
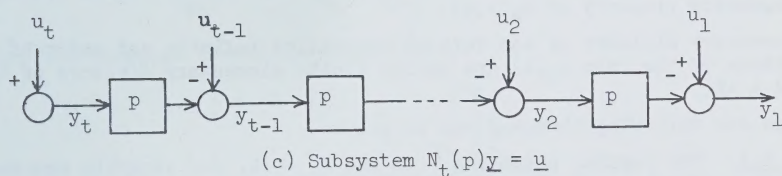
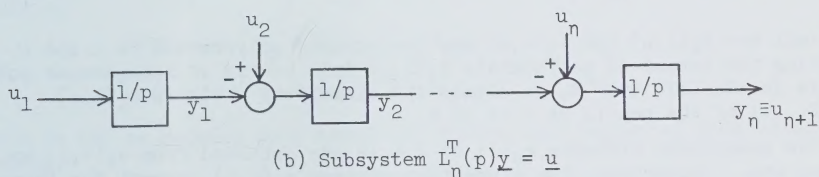
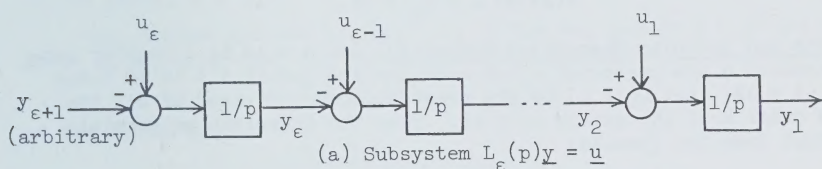


Figure 1.

APPENDIX*

Definition A.1: A binomial $A + \lambda B$ of degree 1 in λ (a scalar) and with coefficients the $m \times n$ rectangular matrices A and B with elements from a number field F , is called a pencil of matrices.

Definition A.2: The rank r of a pencil is equal to the largest of the orders of minors that do not vanish identically in λ .

Definition A.3: A pencil of matrices is called regular if $m=n=r$. In all other cases ($m \neq n$, or $r < m=n$), the pencil is called singular.

Definition A.4: Two pencils of matrices $A + \lambda B$ and $A_1 + \lambda B_1$ of the same dimensions $m \times n$ are called strictly equivalent if there exist nonsingular square matrices P and Q of orders m and n , respectively such that

$$P(A + \lambda B)Q = A_1 + \lambda B_1.$$

Finite and Infinite Elementary Divisors: Let $A + \lambda B$ be a regular pencil. Introduce a new parameter μ and consider the pencil $A + \lambda B$ in terms of μ, λ : $\mu A + \lambda B$. Let $D_k(\lambda, \mu)$ be the greatest common divisor of all the minors of order k of the matrix $\mu A + \lambda B$. Then the invariant polynomials are obtained from the formulas

$$i_1(\lambda, \mu) = \frac{D_n(\lambda, \mu)}{D_{n-1}(\lambda, \mu)}, i_2(\lambda, \mu) = \frac{D_{n-1}(\lambda, \mu)}{D_{n-2}(\lambda, \mu)}, \dots \quad (A.1)$$

where all the $D_k(\lambda, \mu)$ and $i_j(\lambda, \mu)$ are homogeneous polynomials in λ and μ . Splitting the invariant polynomials $i_j(\lambda, \mu)$ into powers of homogeneous polynomials irreducible over F , we obtain the elementary divisors $e_a(\lambda, \mu)$ ($a=1, 2, \dots$) of the pencil $\mu A + \lambda B$ in F .

The elementary divisors $e_a(\lambda)$ of $A + \lambda B$ are obtained from $e_a(\lambda, \mu)$ by setting $\mu=1$. Conversely, the elementary divisor $e_a(\lambda, \mu)$, except for those of the form μ^t , can be obtained from the elementary divisors $e_a(\lambda)$ by using the homogeneity property of $e_a(\lambda, \mu)$.

Elementary divisors of the form μ^t are called infinite and exist if $|B| \neq 0$; these of the form $e_a(\lambda)$ are called finite elementary divisors of the pencil $A + \lambda B$.

Then the following theorems can be proven.

Theorem A.1: Two regular pencils $A + \lambda B$ and $A_1 + \lambda B_1$ are strictly equivalent if they have the same (finite and infinite) elementary divisors.

Theorem A.2: Every regular pencil $A + \lambda B$ can be reduced to a (strictly equivalent) canonical quasi-diagonal form

$$\{N_{t_1}, N_{t_2}, \dots, N_{t_s}, J + \lambda I\} \quad (A.2)$$

where the first s diagonal blocks correspond to infinite elementary divisors $\mu^{t_1}, \mu^{t_2}, \dots, \mu^{t_s}$ of the pencil $A + \lambda B$, are $t_i \times t_i$ square matrices of the form

$$N_{t_i} = \begin{bmatrix} 1 & \lambda & 0 & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & 1 & \lambda \\ 0 & \dots & \dots & \dots & 0 & 1 \end{bmatrix} \quad (A.3)$$

*For details see Reference 3.

and where the normal form of the last block $J + \lambda I$ is uniquely determined by the finite elementary divisors of the given pencil.

Minimal Indices: Let $A + \lambda B$ be an arbitrary singular pencil. Then the equation

$$(A + \lambda B) \underline{x}(\lambda) = \underline{0}$$

admits a fundamental series of polynomial solutions

$$\underline{x}_1(\lambda), \underline{x}_2(\lambda), \dots, \underline{x}_k(\lambda)$$

having the degrees

$$0 \leq \varepsilon_1 \leq \varepsilon_2 \leq \dots \leq \varepsilon_k.$$

The numbers $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_k$ are called the minimal indices for the columns of the pencil $A + \lambda B$.

The minimal indices $\eta_1, \eta_2, \dots, \eta_\ell$ for the row of $A + \lambda B$ are the degrees of a fundamental series of polynomial solutions

$$\underline{y}_1(\lambda), \underline{y}_2(\lambda), \dots, \underline{y}_\ell(\lambda)$$

of the equation

$$(A^T + \lambda B^T) \underline{y}(\lambda) = \underline{0}$$

Theorem A.3: Let the singular pencil $A + \lambda B$ have the complete systems of indices

$$\varepsilon_1 = \dots = \varepsilon_g = 0, \quad 0 < \varepsilon_{g+1} \leq \dots \leq \varepsilon_k$$

$$\eta_1 = \dots = \eta_h = 0, \quad 0 < \eta_{h+1} \leq \dots \leq \eta_\ell$$

then it can be reduced to a (strictly equivalent) canonical quasi-diagonal matrix.

$$\{0; L_{\varepsilon_{g+1}}, \dots, L_{\varepsilon_k}; L_{\eta_{h+1}}^T, L_{\eta_\ell}^T; A_0 + \lambda B_0\} \quad (A.4)$$

where the diagonal blocks have the forms,

$$L_{\varepsilon_{g+j}} = \underbrace{\begin{bmatrix} \lambda & 1 & 0 & \dots & \dots & 0 \\ 0 & \lambda & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & 0 & \lambda & 1 \end{bmatrix}}_{\varepsilon_{g+j}+1} \Bigg\}^{\varepsilon_{g+j}} \quad (A.5)$$

$$L_{\eta_{h+i}} = \underbrace{\begin{bmatrix} \lambda & 1 & \dots & \dots & \dots & 0 \\ 0 & \lambda & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & 0 & \lambda & 1 \end{bmatrix}}_{\eta_{h+i}+1} \Bigg\}^{\eta_{h+i}} \quad (A.6)$$

$A_0 + \lambda B_0$ is a regular pencil and θ is an $h \times g$ zero matrix.

Based on Theorems A.2 and A.3 the following theorems can be proved.

Theorem A.4: (Kronecker): Two arbitrary pencils $A + \lambda B$ and $A_1 + \lambda B_1$ of rectangular matrices of the same dimension $m \times n$ are strictly equivalent if they have the same minimal indices and the same (finite and infinite) elementary divisors.

Theorem A.5: Every arbitrary pencil $A + \lambda B$ can be reduced to a (strictly equivalent) canonical quasi-diagonal form

$$\{\theta, L_{\epsilon_{g+1}}, \dots, L_{\epsilon_k}; L_{\eta_{h+1}}^T, \dots, L_{\eta_\ell}^T; N_{t_1}, \dots, N_{t_s}, J + \lambda I\} \quad (A.7)$$

with the diagonal blocks as have been defined above.

ON THE IDENTIFICATION OF LINEAR TIME-INVARIANT SYSTEMS FROM INPUT-OUTPUT DATA

B. GOPINATH

Bell Telephone Laboratories, Incorporated
Murray Hill, New Jersey

ABSTRACT

This paper presents a new method for computing the parameters which determine the differential equations that govern a linear time invariant system with multiple inputs and outputs. Unlike earlier approaches the present method does not involve computation of the impulse response. One of the main advantages of this method is its easy generalization to the case when the given data is contaminated with noise.

The identification of multiple-input-output linear systems has been a problem of considerable interest during the last few years, because of its importance in circuit and control system theory. In circuit theory the problem is that of synthesizing a linear time invariant circuit to exhibit a prescribed input-output behavior. In control theory, however, the problem arises out of a need to model a given linear system with a suitable set of differential equations, given its input-output behavior. References 1, 2, and 3 deal with the problem of determining the parameters of the differential equation model from the impulse response. To the best of the author's knowledge, there is no published method which determines the impulse response from a finite segment of input-output data in the case of systems with more than one input and output.

1. THE "STATE INVARIANT" DESCRIPTION

In most applications of identification techniques, one is given only a record of input sequences and a record of output sequences, rather than the impulse response function. In these cases it seems best to get an internal description of the system directly from these data; that is, avoid the intermediate step of synthesizing the impulse response. In many applications the structure of systems that are being identified remains the same, while values of parameters change. Therefore, it is convenient to work in a certain coordinate frame which is fixed for the given system. Most important of all, a method of arriving at the values of parameters directly from input/output data is easier to analyze than the method in which impulse response is synthesized, since the sensitivity of intermediate computations required to obtain the impulse response matrix need not be analyzed.

The problem is therefore formulated as follows. Let Σ be a linear system in discrete time modeled by Eqs. (1) and (2).

$$x(s+1) = Fx(s) + Gu(s) \quad (1)$$

$$y(s) = Hx(s) \quad (2)$$

$x(s) \in E^n$ (the "n" dimensional Euclidean space) is the state of Σ at time s, similarly $u(s)$ and $y(s)$ are the m-dimensional input and the p-dimensional output of Σ . F, G, H are real constant matrices of appropriate dimensions. Σ is assumed to be completely reachable and completely observable (for details about these terms see Ref. 4) viz.,

$$\text{Rank of } [G, FG, \dots, F^{n-1}G] = n \quad (3)$$

and

$$\text{Rank of } [H', F'H', \dots, F'^{n-1}H'] = n \quad (4)$$

where prime denotes the transpose. Given a sequence of inputs $u(s)$ and outputs $y(s)$ for $s = 1, 2, \dots, N$ (where N obviously is sufficiently large) find a system $\hat{\Sigma}$ of the same dimension as Σ namely n such that $\hat{\Sigma}$ simulates the input-output behavior of Σ .

Remark 1: It is clear that there are some sequences $u(s)$ which will not be sufficient to uniquely specify $\hat{\Sigma}$. Theorems to be presented later on gives sufficient conditions for $u(s)$ and N which does uniquely determine $\hat{\Sigma}$.

Remark 2: When $\hat{\Sigma}$ is uniquely determined it will be shown that the state of $\hat{\Sigma}$ is uniquely related to the state of Σ . In fact the $\hat{F}$, $\hat{G}$, and $\hat{H}$ of $\hat{\Sigma}$ will be related to the F , G , H of Σ by a nonsingular transformation such that $H\hat{F}^i\hat{G} = HF^iG$ which implies that the impulse responses of Σ and $\hat{\Sigma}$ are identical. Notice that for any nonsingular T

$$\hat{H} = HT^{-1} \quad \hat{F} = TFT^{-1} \quad \hat{G} = TG$$

implies that

$$H\hat{F}^i\hat{G} = HF^iG$$

The main difficulty in obtaining a direct algorithm is in getting at the state $x(s)$ from output sequences when the parameters of the system are not known. When, for example, H in the equation below is identity, or equivalently the output itself is the state, it is easy to find an internal description from sequences of inputs and outputs. For writing this equation as

$$\begin{aligned} x(s+1) &= [F \ G] \begin{bmatrix} x(s) \\ u(s) \end{bmatrix} \\ y(s) &= Hx(s) = x(s), \end{aligned}$$

it follows that given enough observations one can solve for F and G from the above equation for most nontrivial input sequences (see Theorem 2). An easy way is to multiply both sides of this equation by $[x'(s) \ u'(s)]$ and sum from $s = 1$ to $s = N$ where N is the number of observations:

$$\sum_{s=1}^N \{x(s+1)[x'(s) \ u'(s)]\} = [F \ G] \sum_{s=1}^N \left\{ \begin{bmatrix} x(s) \\ u(s) \end{bmatrix} [x'(s) \ u'(s)] \right\}$$

Whenever the matrix multiplying $[F \ G]$ in the above equation has an inverse, there exists a unique solution for F and G .

In the case when $y(s)$ is not the state itself but only a linear function of the state, the problem is much more complex and one has to select certain appropriate components of the output sequence for an external description in terms of the observables, namely $y(i)$ and $u(i)$. The selection of the right components can be done by introducing an operator to be called the selector matrix as defined below.

In describing the theory of the direct identification method, considerable use will be made of the input/output description to be detailed below.

Definition: $\mathcal{S}$ will denote the set of $k \times \ell$ matrices ($k \leq \ell$) with the following properties:

$$1. \ S = \{s_{ij}\} \text{ where } s_{ij} = 0 \text{ or } 1. \quad (5)$$

$$2. \forall i, s_{ij} = 1 \text{ for one and only one } j, \text{ say } j_i. \quad (6)$$

$$3. j_1 < j_2 < \dots < j_k, j_i \leq \ell, i \leq k. \quad (7)$$

Examples of matrices belonging to $\mathcal{S}$ are

$$[1] \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \dots \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ etc.}$$

Any matrix $S \in \mathcal{S}$ will be referred to as a selector matrix, because S operating on a linear space E^ℓ transforms it into a linear space E^k by mapping every vector $x \in E^\ell$ to a vector $y \in E^k$ by selecting the components $j_1, \dots, j_k$ of $x \in E^\ell$.

The description that will be presented is an "external" description in the sense that the dynamical equations are given in terms of quantities which can be observed from outside; i.e., values of input and values of output.

Consider a completely reachable and completely observable discrete time system Σ represented as follows:

$$x(s+1) = Fx(s) + Gu(s), \quad (8)$$

$$y(s) = Hx(s), \quad H: p \times n; \quad F: n \times n; \quad G: n \times m. \quad (9)$$

Completely observable implies

$$\rho([F':H'])^* = n. \quad (10)$$

$\rho(A)$ = rank of A . Therefore, $\exists$ an $S \in \mathcal{S}$, such that

$$S \begin{bmatrix} H \\ \vdots \\ HF^{n-1} \end{bmatrix} = T \text{ where } T \text{ is nonsingular;} \quad (11)$$

i.e., T^{-1} exists. Without loss of generality it can be assumed from Remark 2 that, $T = I$ so far as the external description is concerned. Using Eqs. (8) and (9) repeatedly, it follows that

$$y(s) = Hx(s),$$

$$y(s+1) = Hx(s+1) = HFx(s) + Gu(s), \quad (12)$$

$$y(s+n-1) = HF^{n-1}x(s) + HF^{n-2}Gu(s) + \dots + HGu(s+n-2).$$

Let

$$\bar{y}'(s) \triangleq [y'(s) \quad y'(s+1) \quad \dots \quad y'(s+n-1)]. \quad (13)$$

$$\bar{u}'(s) \triangleq [u'(s) \quad u'(s+1) \quad \dots \quad u'(s+n-1)]. \quad (14)$$

Then, writing Eqs. (12) in vector form, and also using Eqs. (13) and (14), it follows that

$$\bar{y}'(s) = \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{bmatrix} x(s) + \begin{bmatrix} 0 & 0 & 0 \\ HG & 0 & 0 \\ \vdots & \vdots & \vdots \\ HF^{n-2}G & HF^{n-1}G & \dots 0 \end{bmatrix} \bar{u}'(s). \quad (15)$$

Let

$$[F':H'] \triangleq [H'F'H', \dots, F'^{(n-1)}H].$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ HG & 0 & 0 & 0 & 0 & 0 \\ HFG & HG & 0 & 0 & 0 & 0 \\ \cdot & HFG & HG & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ HF^{n-3}G & HF^{n-4}G & HF^{n-5}G & \dots & HG & 0 & 0 \\ HF^{n-2}G & HF^{n-3}G & HF^{n-4}G & \dots & HFG & HG & 0 \end{bmatrix} \triangleq R_1;^* \quad (16)$$

then multiplying both sides of Eq. (15) by S, using the comments given below Eq. (11), it follows that

$$S\bar{y}(s) = x(s) + SR_1\bar{u}(s). \quad (17)$$

Once again, using Eq. (9),

$$x(s+1) = Fx(s) + Gu(s),$$

which because of Eq. (17) with s replaced by s + 1 reduces to

$$x(s+1) = S\bar{y}(s+1) - SR_1\bar{u}(s+1), \quad (18)$$

and substituting Eq. (9) for x(s) in Eq. (17) gives

$$S\bar{y}(s+1) = F(S\bar{y}(s) - SR_1\bar{u}(s)) + S \begin{bmatrix} H \\ HF \\ \cdot \\ \cdot \\ \cdot \\ HF^{n-1} \end{bmatrix} Gu(s) + SR_1\bar{u}(s+1). \quad (19)$$

Since it has been shown that [see Eq. (11)]:

$$S \begin{bmatrix} H \\ HF \\ \cdot \\ \cdot \\ \cdot \\ HF^{n-1} \end{bmatrix} = I, \quad (20)$$

it follows that

$$S\bar{y}(s+1) = FS\bar{y}(s) + R\bar{u}(s), \quad (21)$$

where[†]

$$R \triangleq -FSR_1 + S \begin{bmatrix} HG & 0 & 0 & 0 & 0 \\ HFG & HG & 0 & 0 & 0 \\ HF^2G & HFG & 0 & 0 & 0 \\ \cdot & \cdot & 0 & \cdot & \cdot \\ HF^{n-1}G & HF^{n-2}G & \cdot & \cdot & HG \end{bmatrix}. \quad (22)$$

Equation (21) gives a relation between the input sequence u(i) and the output sequence y(i) which does not involve the state. It is an external description in the sense that the variables in Eq. (21), namely u(i) and y(i), can be measured externally.

* The last column of zeroes in R₁ is added so that $\bar{y}$ and $\bar{u}$ may be consistently defined.

† In adding SR₁ $\bar{u}(s+1)$ to the second term in Eq. (20), the last column of R₁ may be dropped because it is all zeroes.

From Eq. (22) it follows that if R is partitioned as

$$[R_0 \ R_1 \ \dots \ R_{n-1}], \quad R_i: n \times m \quad \forall i, \quad (23)$$

then

$$R_{n-1} = S \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ HG \end{bmatrix} \quad (24)$$

It is obvious how one obtains the columns of the second product in Eq. (22). To obtain the contribution from $-FSR_1$, notice from Eq. (16) that S times the second column from the end of R_1 is from Eq. (24) merely R_{n-1} . Therefore, the second column of FSR_1 from the end is simply FR_{n-1} and therefore

$$R_{n-2} = -FR_{n-1} + S \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ HG \\ HFG \end{bmatrix}. \quad (25)$$

Now notice that $R_{n-2} + FR_{n-1}$ is the S times third column from the end of R_1 . Therefore the third column from the end of R is

$$R_{n-3} = -FR_{n-2} - F^2R_{n-1} + S \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \\ HG \\ HFG \\ HF^2G \end{bmatrix}$$

Continuing in the same vein,

$$R_{n-4} = -FR_{n-3} - F^2R_{n-2} - F^3R_{n-1} + S \begin{bmatrix} 0 \\ \cdot \\ 0 \\ HG \\ HFG \\ HF^2G \\ HF^3G \end{bmatrix}$$

and finally

$$S \begin{bmatrix} HG \\ HFG \\ \cdot \\ \cdot \\ \cdot \\ HF^{n-1}G \end{bmatrix} = R_0 + FR_1 + \dots + F^{n-1}R_{n-1}. \quad (26)$$

Now, since it was possible to choose a basis such that

$$S \begin{bmatrix} HG \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ HF^{n-1}G \end{bmatrix} = IG,$$

one has

$$G = R_0 + FR_1 + \dots + F^{n-1}R_{n-1}. \quad (27)$$

Equation (21) may be written in the form

$$S\bar{y}(s+1) = [F \ R] \begin{bmatrix} S\bar{y}(s) \\ \bar{u}(s) \end{bmatrix}$$

and may in principle be solved for F and R. Thus from Eqs. (27) it is clear that if the values for $u(i)$ and $y(i)$ were given and S were known, one could also solve for one set of values for G; and since in most cases H is full rank, H can be assumed to be $[I \ 0]$.

2. THE MINIMAL REPRESENTATION AND THE DIRECT ALGORITHM

It was shown in Section 1 that, corresponding to every internal description of Σ which is completely controllable and completely observable, there is a description in the form of Eq. (21). In this section it will be shown that from the knowledge of the values of $u(i)$, $i = 1, \dots, N$, and $y(i)$, $i = 1, \dots, N$, one can get the internal description of Σ under very general conditions on $u(i)$.

Central to the discussion are a few results which will be presented in the form of theorems for the sake of clarity and precision.

Since one of the most important steps in realizing a system is to find the minimum dimension of that system, we will prove two theorems which can be used to find the minimal n.

Given $u(i)$, $i = 1, \dots, N$, the inputs to a system Σ of dimension n which is completely observable and completely reachable, and the corresponding outputs $y(i)$, $i = 1, \dots, N$, the following propositions hold true:

Note 1: It will be assumed in the following that the column dimension k of the selector matrix is always a multiple of p; further, if $k = rp$, then

$$(r-1)p < j_l \leq rp.$$

It is obvious that there is no loss of generality involved in this assumption. (l here is the nondimension of S.)

Note 2: In the definition of $\bar{y}(s)$ and $\bar{u}(s)$ in Eq. (13) and Eq. (14), the n should be replaced by r defined in Note 1 above.

Theorem 1: Let S be $\ell \times k (= rp)$ then

$$\rho \left\{ S \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{r-1} \end{bmatrix} \right\} < \ell \quad (28)$$

implies that

$$\sum_{s=1}^{N-r+1} \begin{bmatrix} S\bar{y}(s) \\ \bar{u}(s) \end{bmatrix} [\bar{y}'(s)S' \quad \bar{u}'(s)] \text{ is a singular matrix} \quad (29)$$

for every sequence $u(i)$, $i = 1, 2, \dots, N$.

Proof: Multiplying Eq. (10) on the left by S and replacing n by r we have,

$$S\bar{y}(s) = S \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{r-1} \end{bmatrix} x(s) + SR_1 \bar{u}(s) \quad (30)$$

Because of Eq. (28) $\exists$ a vector $z \neq 0$ and in E^l such that

$$z'S \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{r-1} \end{bmatrix} = 0 \quad (31)$$

Therefore, multiplying Eqs. (29) on the left by z' gives

$$z'S\bar{y}(s) = z'SR_1\bar{u}(s) \quad (32)$$

Therefore,

$$[z' \quad -z'SR_1] \begin{bmatrix} \bar{y}(s) \\ \bar{u}(s) \end{bmatrix} = 0 \quad \forall s \leq N - r + 1 \quad (33)$$

which implies that

$$\sum_{s=1}^{N-r+1} \begin{bmatrix} \bar{y}(s) \\ \bar{u}(s) \end{bmatrix} [\bar{y}(s)S' \quad \bar{u}'(s)] \text{ is singular.} \quad \text{Q.E.D.}$$

Theorem 2: If Σ is completely observable and completely reachable the matrices F , G , and H are $n \times n$, $n \times m$, and $p \times n$ respectively; then $\exists$ an $S: n \times p$ such that

$$T \triangleq \sum_{s=1}^{n+nm} \begin{bmatrix} \bar{y}(s) \\ \bar{u}(s) \end{bmatrix} [\bar{y}'(s)S' \quad \bar{u}'(s)] > 0 \text{ almost surely} \quad (34)$$

where $u(i)$ are random variables having a joint nonlattice distribution.*

Proof: The first step in the proof consists of establishing Lemma 1.

Lemma 1: $T > 0$ if and only if

$$\sum_{s=1}^{n+nm} \begin{bmatrix} x(s) \\ \bar{u}(s) \end{bmatrix} [x'(s) \quad \bar{u}'(s)] > 0 \quad (35)$$

Proof of Lemma 1: If $T \not> 0$, $\exists z'$ such that $z' \triangleq [z'_1 \quad z'_2] \neq 0$, and

$$z'_1 S\bar{y}(s) + z'_2 \bar{u}(s) = 0 \quad \forall s. \quad (36)$$

Since Σ is completely observable, multiplying Eq. (17)

$$S\bar{y}(s) = x(s) + SR_1\bar{u}(s) \quad (37)$$

on the left by z'_1 one obtains

$$z'_1 S\bar{y}(s) = z'_1 x(s) + z'_1 SR_1\bar{u}(s). \quad (38)$$

Combining Eqs. (36) and (38),

$$z'_1 x(s) + (z'_1 SR_1 - z'_2)\bar{u}(s) = 0 \quad \forall s, \quad (39)$$

and

$$[z'_1, z'_1 SR_1 - z'_2] \neq 0;$$

* A nonlattice distribution is one in which no nonzero probability mass is concentrated on a surface less than the dimension of the random variable.

for if $[z'_1, z'_1 SR_1 - z'_2] = 0$, then $[z'_1 \ z'_2] = 0$, which contradicts $z \neq 0$. Therefore,

$$\sum_{s=1}^{n+nm} \begin{bmatrix} x(s) \\ \bar{u}(s) \end{bmatrix} [x'(s) \ \bar{u}'(s)] \neq 0. \quad (40)$$

Now suppose $T > 0$. Let

$$\sum_{s=1}^{n+nm} \begin{bmatrix} x(s) \\ u(s) \end{bmatrix} [x'(s) \ \bar{u}(s)] \neq 0.$$

Then $\exists$ a $z' = [z'_1 \ z'_2] \neq 0$ such that

$$z'_1 x(s) + z'_2 \bar{u}(s) = 0 \quad \forall s. \quad (41)$$

Again multiplying Eq. (37) by z'_1 and using Eq. (41), it follows that

$$z'_1 S \bar{y}(s) = -z'_2 \bar{u}(s) + z'_1 SR_1 \bar{u}(s) \quad (42)$$

$$z'_1 S \bar{y}(s) + (z'_2 - z'_1 SR_1) \bar{u}(s) = 0 \quad \forall s. \quad (43)$$

Once again $[z'_1, (z'_2 - z'_1 SR_1)] \neq 0$ since $z \neq 0$, which contradicts $T > 0$.

The proof of Lemma 2 will now complete the proof of Theorem 2.

Lemma 2: If (1) Σ is completely controllable, (2) $u(i)$ are random variables with a joint nonlattice distribution, then the $(n+nm) \times (n+nm)$ matrix

$$\begin{bmatrix} x(1) & \dots & x(n+nm) \\ u(1) & \dots & u(n+nm) \end{bmatrix} \quad (44)$$

is nonsingular almost surely.

Proof: From Lemma A.2 in Appendix A (Ref. 5) it follows that if

$$z(s+1) = F_1 z(s) + G_1 u(s), \quad (45)$$

$$\text{with } F_1(n+nm) \times (n+nm),$$

then $[z(1), \dots, z(n+nm)]$ is nonsingular with probability one, if F_1, G_1 is completely controllable. Further, from Eqs. (8) and the definition of $\bar{u}$, it is clear that

$$\begin{bmatrix} x(s+1) \\ u(s+1) \\ \vdots \\ u(s+n) \end{bmatrix} = \begin{bmatrix} F & G & 0 & \dots & \dots & 0 \\ 0 & 0 & I & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \vdots & \dots & \dots & I \\ 0 & 0 & \vdots & \dots & \dots & 0 \end{bmatrix} \begin{bmatrix} x(s) \\ u(s) \\ \vdots \\ u(s+n-1) \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ I \\ \vdots \end{bmatrix} u(s+n) \quad (46)$$

Therefore, identifying F_1 and G_1 as

$$\begin{bmatrix} F & G & 0 & \dots & \dots & 0 \\ 0 & 0 & I & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \dots & \dots & \vdots \\ 0 & 0 & \vdots & \dots & \dots & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ I \\ \vdots \end{bmatrix}$$

respectively, Eq. (44) follows since it can easily be shown that $[F \ G]$ controllable implies that $[F_1 \ G_1]$ is completely controllable.

Lemma 2 implies that the matrix in Eq. (40) is positive definite since in general A nonsingular $\rightarrow A^T A > 0$, which implies (34) by 1.

3. THE COMPUTATIONAL METHOD

The main part of the algorithm as would be expected from the above discussion is to determine the right selector matrix. Once this has been done it is easy to solve for the parameters. In order to utilize certain properties of the matrix $[F':H']$, a class of matrices $\bar{S} \subset S$ is detailed below, since $\bar{S} \subset S$ the number of different selector matrices one has to try is smaller than S .

Definition: $\bar{S}$ is the set of matrices $S \in S$ such that if S is $k \times l$, then

1. k is an integral multiple of p and further, if $k = rp$, then

$$(r-1)p < j_l \leq rp$$

where j_l is as defined in Eq. (5).

2. $j_1 - j_{i-1} \leq 2p$.

Observe that 1) there always exists an $S \in \bar{S}$ such that Eq. (11) holds, since as can easily be proved, if

$$\rho([H', F'H', \dots, F'^S H']) = \rho([H', F'H', \dots, F'^{(s+1)} H']) = q$$

then

$$\rho([H', F'H', F'^{(s+j)} H']) = q \quad j = 0, 1, 2, \dots$$

so that in spite of condition 2) in the above definition, there exists an $S \in \bar{S}$ such that Eq. (11) holds.

3. The formulas (21) and (27) are still valid for any $S \in \bar{S}$ satisfying Eq. (11), with n replaced everywhere by r defined in condition 1) in the definition of $\bar{S}$ above.

Now from Theorems 1 and 2 and the above discussion, the direct algorithm can be summarized as follows.

It can be assumed without loss of generality that: (1) $N \geq n + (m+1)n$; i.e., there is sufficient number of observations to determine the internal description uniquely. n here is the minimal dimension of the system to be identified. (2) H has full rank. (3) $n \geq p$.

Step 1: Since $n < N/(m+2)$, let $\bar{N}$ be the largest integer $\leq N/(m+2)$. Then $n < \bar{N}$. In order to arrive at the right S , one starts with an $S \in \bar{S}$ of row dimension $\bar{N}$ and tests the nonsingularity of

$$T \triangleq \sum_{s=1}^{(m+1)\bar{N}} \begin{bmatrix} S\bar{y}(s) \\ \bar{u}(s) \end{bmatrix} [\bar{y}'(s)S' \quad \bar{u}'(s)]$$

for all $S \in \bar{S}$ and having row dimension $\bar{N}$. If T is nonsingular, $\bar{N} = n$. If T is singular, then reduce the row dimension of S by 1 and repeat the test. Repeat the procedure until T becomes nonsingular. The row dimension of S will then be n and let r be as defined in condition (1) in the definition of $\bar{S}$; i.e., S is $n \times rp$.

Step 2: Solve for F, R as follows.

$$[F \ R] = \left\{ \sum_{s=1}^{(m+1)\bar{N}} S y(s+1) [\bar{y}(s) S' \quad \bar{u}'(s)] \right\} T^{-1}$$

Step 3: Solve for G from the following formula.

$$G = R_0 + F R_1 + \dots + F^{r-1} R_{r-1},$$

where $S: n \times r$ and R_i are the partitions of R such that

$$R = [R_0 \ R_1 \ \dots \ R_{r-1}]$$

and $R_i = n \times m \ \forall i, \dots, r-1$. H can be assumed to be $[I \ 0]$ where the identity has dimension p .

In the case when Σ is a continuous-time system, the algorithm presented above applies with appropriate modifications. In the definitions of $\bar{y}(s)$ and $\bar{u}(s)$, s now assumes values in $\mathbb{R}$ and $y(s+1)$ should be replaced by $y^{(i)}(s)$, denoting the i th derivative of $y(s)$ evaluated at s . The summation signs should be replaced by integration over an interval. The formulas for the parameters become

$$[F \ R] = \left\{ \int_t^{t+\varepsilon} S \bar{y}^{(\cdot)}(s) [\bar{y}'(s) S' \quad \bar{u}'(s)] ds \right\} T^{-1}$$

where

$$T = \int_t^{t+\varepsilon} \begin{bmatrix} S \bar{y}(s) \\ \bar{u}(s) \end{bmatrix} [\bar{y}'(s) S' \quad \bar{u}'(s)] ds.$$

G can be obtained from R exactly as in the above algorithm for the discrete time case.

In the case when observations are contaminated with noise, this method can be generalized to yield consistent estimates for the parameters. This however will be presented elsewhere.

REFERENCES

1. B. L. Ho, "On Effective Construction of Realizations from Input-Output Description," Ph.D. Dissertation, 1965, Department of Engineering Mechanics, Stanford University, California.
2. B. L. Ho and R. E. Kalman, "Effective Construction of Linear State-Variable Models from Input/Output Data," Proceedings of the Third Annual Allerton Conference on Circuit and System Theory, 1965.
3. D. C. Youla, "The Synthesis of Linear Dynamical Systems from Prescribed Weighting Patterns," J. SIAM, Vol. 14, May 1966.
4. R. E. Kalman, "Mathematical Description of Linear Systems," SIAM Journal on CONTROL, Vol. 1, No. 1.
5. B. Gopinath, "On the Identification and Control of Linear Systems," Ph.D. Dissertation Submitted to Department of Electrical Engineering, Stanford University, California, 1968.

A CONTINUOUS STOCHASTIC APPROXIMATION ALGORITHM FOR NON-LINEAR SYSTEM IDENTIFICATION

D.W. RICKER
Electrical Engineering
Purdue University
Lafayette, Indiana

G.N. SARIDIS
Electrical Engineering
Purdue University
Lafayette, Indiana

This paper presents two stochastic approximation algorithms for the identification of non-linear continuous forced systems in the presence of additive noise. It is assumed that noisy measurements of the system states are available and some knowledge of the non-linearities exists. The additive noise components are assumed to be uncorrelated with bounded variance and zero mean. Additionally, the noise is assumed to be white with respect to the system dynamics.

A mean square state error

$$E \{J\} = E \{\|X - \hat{X}\|^2\} \quad (1)$$

is found where X represents the system state vector and $\hat{X}$ represents a vector of model states. Since it is assumed that the vector X is only available from noisy measurements, the error (1) takes the form

$$E \{J_z\} = E \{\|Z - \hat{X}\|^2\} \quad (2)$$

where Z represents the vector of noisy state measurements. Under the assumptions given concerning the noisy components it can be shown that a minimization of (2) is equivalent to a minimization of (1).

The model system has the same dimensions as that of the plant and the non-linearities are approximated by a function of the form

$$\alpha(\hat{X}) = (C^T, \phi(\hat{X})) \quad (3)$$

where C represents a vector of coefficients to be determined by the identification procedure and $\phi(\hat{X})$ represents a vector of independent function of the states $\hat{X}$. For example, in the case of continuous non-linearities the function (3) may take the form

$$\alpha(\hat{X}) = c_1 + c_2 \hat{x} + c_3 \hat{x}^2 + \dots + c_N \hat{x}^{N-1}. \quad (4)$$

Two computational algorithms for the estimation of the constants c minimizing (1) are proposed. The first is a continuous Robbins-Monro technique originally proposed by Sakrison where the parameters are estimated by

$$\frac{dc}{dt} = -\gamma(t) \frac{\partial J}{\partial c} \quad (5)$$

where $\gamma(t)$ is a function meeting certain conditions necessary for the convergence of the algorithm. In order to implement (5) it

is also necessary to generate the sensitivity functions of the model since the parameters C are not explicit in (1). For systems involving a large number of parameters this will involve a significant amount of analog computation. In order to circumvent this problem, a second algorithm is proposed which is a modification of the classical Kiefer-Wolfowitz procedure in which the gradient $\frac{\partial J_z}{\partial C}$ is estimated by calculating the error J_z

at two points and thus approximating the derivative for each component of the gradient. Additionally, the parameters are estimated sequentially which allows the use of one model system for all of the parameters resulting in a significant reduction in the amount of computational hardware.

Results are presented for a non-linear system characterized by

$$\dot{x}(t) + .5(1+x(t)^2) x(t) = U \sin w t \quad (6)$$

using a model of the form

$$\dot{\hat{x}}(t) + (c_1 \vartheta_1 + c_2 \vartheta_2 + c_3 \vartheta_3) \hat{x}(t) = U \sin wt \quad (7)$$

where $\vartheta_1 = 1$, $\vartheta_2 = \hat{x}(t)$ and $\vartheta_3 = \hat{x}^2(t)$. A correct identification should thus result in $c_1 = .5$, $c_2 = 0$, $c_3 = .5$ which is the result obtained.

ROUND-OFF ERROR OF FLOATING-POINT DIGITAL FILTERS

T. KANEKO and B. LIU
Princeton University
Princeton, New Jersey

ABSTRACT

This paper is concerned with the accumulation of round-off error in a floating-point digital filter. The error committed at each arithmetic operation is assumed to be an independent random variable uniformly distributed in $(-2^{-t}, 2^{-t})$ where t is the length of the mantissa. Expression for the mean square error is derived and a numerical example is given.

INTRODUCTION

Consider a digital filter specified by the input-output relationship:

$$w_n = \sum_{k=0}^M b_k x_{n-k} - \sum_{k=1}^N a_k w_{n-k} \quad (1)$$

where $\{x_n\}$ is the input sequence and $\{w_n\}$ is the output sequence. There are three common sources of errors associated with the filter of Eq. (1), namely

- (1) quantization error - caused by the quantization of the input signal into steps of finite range,
- (2) coefficient accuracy - caused by the fact that the coefficients a_k and b_k are realized with finite word length,
- (3) round-off accumulation - caused by the accumulation of errors committed at each arithmetic operation because these operations are carried out with only finite bit accuracy.

The actual computed output reference is therefore in general different from $\{w_n\}$. We shall denote the actual output by $\{y_n\}$ and we shall call $\{w_n\}$ the ideal output sequence.

When the filter of Eq. (1) is realized with fixed-point arithmetic, the above error problems have been studied extensively [1]. Recently, Sandberg [2] studied the error problem for floating-point arithmetic and derived an absolute upper bound of the error accumulation due to round-off. This paper derives expression of the mean squared error caused by the round-off accumulation.

Throughout this paper we shall be concerned with binary machines. Thus, a floating point number is stored in the form of $(\text{sgn}) \cdot a \cdot 2^b$ where a and b are binary numbers each of fixed number of bits. The number a is called the mantissa and b the exponent. We shall assume that enough bits are allowed to the exponent b so that no computed number will lie outside the permissible range. We shall use the notation $\text{fl}(\cdot)$ to denote the machine number resulted from the arithmetic operation specified by the parenthesis in some designated order. Thus

$fl(x+y)$ is the calculated sum of $x+y$, and $fl(ax+by)$ is the calculated sum of two terms; one is the calculated product of a and x , the other is the calculated product of b and y . It is known [3] that

$$fl(x+y) = (x+y)(1+\epsilon) \quad (2)$$

with¹

$$|\epsilon| \leq 2^{-t}$$

where t is the number of bits of the mantissa. Also,

$$fl(xy) = xy(1+\epsilon) \quad (3)$$

Again with

$$|\epsilon| \leq 2^{-t}$$

That is, for each addition or multiplication, an error is committed which is proportional to the ideal results obtained with infinite precision. We shall assume that all numbers a_n , b_n , x_n are machine numbers.

CALCULATION OF ACTUAL OUTPUT SEQUENCE

To illustrate the approach of this paper, consider a second order filter specified by

$$w_n = b_0 x_n - (a_1 w_{n-1} + a_2 w_{n-2})$$

The actual computed output is

$$y_n = fl[b_0 x_n - (a_1 y_{n-1} + a_2 y_{n-2})]$$

The calculation is to be performed in the following manner. First, the products $a_1 y_{n-1}$, $a_2 y_{n-2}$, and $b_0 x_n$ are calculated separately. $a_1 y_{n-1}$ and $a_2 y_{n-2}$ are added next. Finally, this sum is subtracted from $b_0 x_n$ to obtain y_n .

Following Sandberg [2], a flow diagram of Figure 1² may be drawn by using Eqs. (2) and (3).

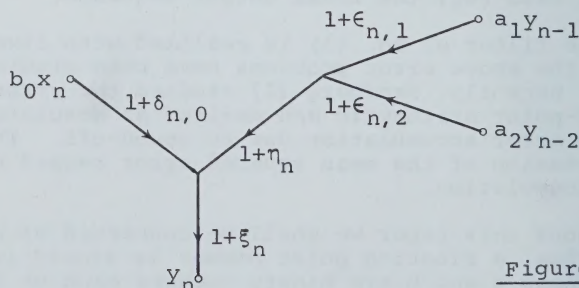


Figure 1

¹It is assumed here that the accumulator is of double precision. For single precision accumulators, slight modification is necessary.

²Slight modifications are necessary when some of the coefficients are one or zero.

The quantities $\delta_{n,0}$, $\epsilon_{n,1}$, $\epsilon_{n,2}$, η_n , ξ_n are all bounded in absolute value by 2^{-t} . Since these are errors caused by round-off at each arithmetic step, we may assume that they are independent random variables uniformly distributed in $(-2^{-t}, 2^{-t})$. Therefore the actual output $\{y_n\}$ is seen to be given explicitly by

$$y_n = b_0 \theta_{n,0} x_n - \sum_{k=1}^2 a_k \phi_{n,k} y_{n-k}$$

where:

$$\theta_{n,0} = (1+\delta_{n,0})(1+\xi_n)$$

$$\phi_{n,1} = (1+\epsilon_{n,1})(1+\eta_n)(1+\xi_n)$$

$$\phi_{n,2} = (1+\epsilon_{n,2})(1+\eta_n)(1+\xi_n)$$

In general, for the digital filter of Eq. (1), we have

$$y_n = fl\left(\sum_{k=0}^M a_k x_{n-k} - \sum_{k=1}^N b_k y_{n-k}\right)$$

and the corresponding flow graph showing the effect of round-off error is given by Figure 2.

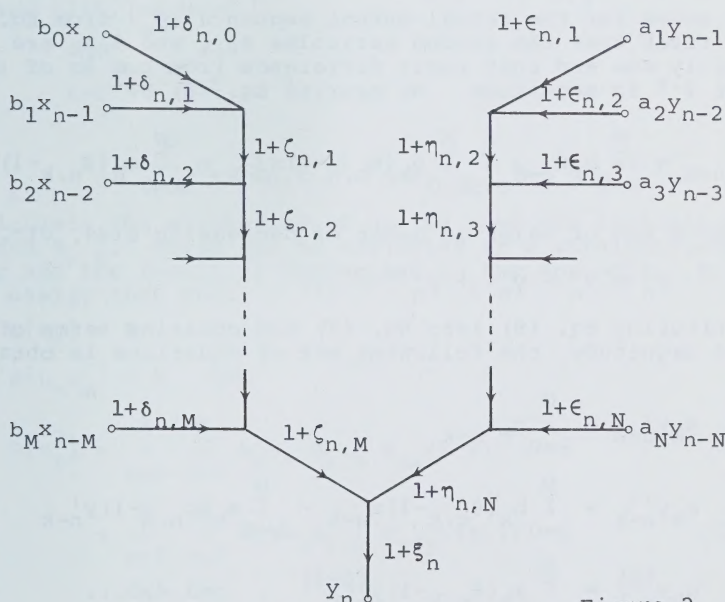


Figure 2

³For single precision accumulators, a slight modification is needed.

The actual output sequence is therefore given by

$$y_n = \sum_{k=0}^M b_k \theta_{n,k} x_{n-k} - \sum_{k=1}^N a_k \phi_{n,k} y_{n-k} \quad (4)$$

where

$$\begin{aligned} \theta_{n,0} &= (1+\xi_n)(1+\delta_{n,0}) \prod_{i=1}^M (1+\zeta_{n,i}) \\ \theta_{n,j} &= (1+\xi_n)(1+\delta_{n,j}) \prod_{i=j}^M (1+\zeta_{n,i}) \quad j=1,2,\dots,M \\ \phi_{n,1} &= (1+\xi_n)(1+\epsilon_{n,1}) \prod_{i=2}^N (1+\eta_{n,i}) \\ \phi_{n,j} &= (1+\xi_n)(1+\epsilon_{n,j}) \prod_{i=j}^N (1+\eta_{n,i}) \quad j=2,3,\dots,N \end{aligned} \quad (5)$$

By defining $a_0 = 1$, $\phi_{n,0} = 1$, we may write Eq. (4) as

$$\sum_{k=0}^N a_k \phi_{n,k} y_{n-k} = \sum_{k=0}^M b_k \theta_{n,k} x_{n-k} \quad (6)$$

where ξ_n , $\delta_{n,k}$, $\zeta_{n,k}$, $\epsilon_{n,k}$, $\eta_{n,k}$ are independent random variables, each uniformly distributed in $(-2^{-t}, 2^{-t})$.

To solve for the actual output sequence $\{y_n\}$ from Eq. (6), we note first that the random variables $\phi_{n,k}$ and $\theta_{n,k}$ are essentially one and that their difference from one is of the order of 2^{-t} in magnitude. We rewrite Eq. (6) as

$$\sum_{k=0}^N a_k y_{n-k} = \sum_{k=0}^M b_k x_{n-k} + \sum_{k=0}^M b_k (\theta_{n,k}-1) x_{n-k} - \sum_{k=0}^N a_k (\phi_{n,k}-1) y_{n-k} \quad (7)$$

and y_n as a sum of terms in order of decreasing size, viz.,

$$y_n = y_n' + y_n'' + y_n''' + \dots \quad (8)$$

On substituting Eq. (8) into Eq. (7) and equating terms of like order of magnitude, the following set of equations is obtained.

$$\sum_{k=0}^N a_k y_{n-k}' = \sum_{k=0}^M b_k x_{n-k} \quad (9)$$

$$\sum_{k=0}^N a_k y_{n-k}'' = \sum_{k=0}^M b_k (\theta_{n,k}-1) x_{n-k} - \sum_{k=0}^N a_k (\phi_{n,k}-1) y_{n-k}' \quad (10)$$

$$\sum_{k=0}^N a_k y_{n-k}^{(p)} = \sum_{k=0}^N a_k (\theta_{n,k}-1) y_{n-k}^{(p-1)}, \quad p=3,4,5,\dots \quad (11)$$

It is to be expected that the order of magnitude of $y_n^{(p)}$ is smaller than that of $y_n^{(p-1)}$ by a ratio of roughly 2^{-t} . This can indeed be shown to be true. Therefore we will only be interested in the y_n' and the y_n'' sequences. A comparison of Eq. (9) with Eq. (1) shows that $\{y_n'\}$ is identical with the

$\{w_n\}$ sequence, the ideal output. We may thus identify y_n'' with the error due to round-off error and denote it by e_n . That is,

$$e_n = y_n - w_n = y_n'' \quad (12)$$

STATISTICS OF ERROR SEQUENCES $\{e_n\}$

Assume the input $\{x_n\}$ is zero mean, and wide-sense stationary (w.s.s) with autocorrelation function $R_{xx}(n)$ and power spectral density $\Phi_{xx}(z)$. Then it is clear from Eq. (9) that y_n' or w_n is zero mean and w.s.s. Its power spectral density $\Phi_{y'y'}(z)$ is related to that of $\{x_n\}$ by

$$\Phi_{y'y'}(z) = \frac{N(z)N(1/z)}{D(z)D(1/z)} \Phi_{xx}(z) \quad (13)$$

and its autocorrelation function $R_{y'y'}(n)$ is given by⁴

$$R_{y'y'}(n) = \frac{1}{2\pi j} \oint \Phi_{y'y'}(z) z^n \frac{dz}{z} \quad (14)$$

where

$$N(z) = \sum_{k=0}^M b_k z^{-k}, \text{ and } D(z) = \sum_{k=0}^N a_k z^{-k}.$$

Furthermore, y_n' is independent of $\theta_{m,j}$ and $\phi_{m,j}$. From Eqs. (10) and (12), we have

$$\sum_{k=0}^N a_k e_{n-k} = u_n \quad (15)$$

with

$$u_n = \sum_{k=0}^N b_k (\theta_{n,k-1}) x_{n-k} - \sum_{k=0}^N a_k (\phi_{n,k-1}) y'_{n-k} \quad (16)$$

To calculate the statistics of u_n , we need the statistics of $\theta_{n,k}$ and $\phi_{n,k}$. These can be evaluated in a straightforward manner and the result is summarized in the Appendix. It can be shown easily then that

$$E\{u_n\} = 0 \quad (17)$$

$$E\{u_n u_m\} = 0 \quad n \neq m \quad (18)$$

$$\begin{aligned} E\{u_n^2\} &= \sum_{k=0}^M \sum_{i=0}^M b_k b_i B_{k,i} R_{xx}(k-i) \\ &+ \sum_{k=1}^N \sum_{i=1}^N a_k a_i A_{k,i} R_{y'y'}(k-i) \\ &- 2 \sum_{k=0}^M \sum_{i=1}^N b_k a_i q R_{xy}(k-i) \end{aligned} \quad (19)$$

where $q = 2^{-2t/3}$ is the variance of a random variable uniformly

⁴ Unless otherwise specified, contour integration in the z -plane shall be along $|z| = 1$.

distributed in $(-2^{-t}, 2^{-t})$, $R_{xy}(k-i)$ is the cross correlation function between $\{x_n\}$ and $\{y_n\}$, $A_{k,i}$ and $B_{k,i}$ are given by:

$$A_{k,i} \triangleq E\{(\phi_{n,k-1})(\phi_{n,i-1})\} \\ = \begin{cases} (1+q)^{N+2-\max(k,i)} - 1 & k \neq i \text{ or } k=i=1 \\ (1+q)^{N+3-k} - 1 & k=i \neq 1 \end{cases} \quad (20)$$

$$B_{k,i} \triangleq E\{(\theta_{n,k-1})(\theta_{n,i-1})\} \\ = \begin{cases} (1+q)^{M+2-\max(k,i)} - 1 & k \neq i \text{ or } k=i=0 \\ (1+q)^{M+3-k} - 1 & k=i \neq 0 \end{cases} \quad (21)$$

Thus we see $\{u_n\}$ is white and w.s.s. with variance of u_n given by Eq. (19) which may be rewritten as

$$E\{u_n^2\} = \frac{1}{2\pi j} \oint [|B(z)|^2 \Phi_{xx}(z) + |A(z)|^2 \Phi_{yy}(z) - 2C(z) \Phi_{xy}(z)] \frac{dz}{z} \quad (22)$$

where

$$|B(z)|^2 = \sum_{k=0}^M \sum_{i=0}^M b_k b_i B_{k,i} z^{k-i} \\ |A(z)|^2 = \sum_{k=1}^N \sum_{i=1}^N a_k a_i A_{k,i} z^{k-i} \\ C(z) = q \sum_{k=0}^M \sum_{i=1}^N b_k a_i z^{k-i}$$

By using Eq. (13) and the relationship

$$\Phi_{xy}(z) = \frac{N(z)}{D(z)} \Phi_{xx}(z)$$

we have

$$E\{u_n^2\} = \frac{1}{2\pi j} \oint [|B(z)|^2 + |A(z)|^2 \frac{N(z)}{D(z)} - 2C(z) \frac{N(z)}{D(z)}] \Phi_{xx}(z) \frac{dz}{z} \quad (23)$$

From Eq. (15), we see that e_n has zero mean and is w.s.s. Its power spectral density $\Phi_{ee}(z)$ is given by

$$\Phi_{ee}(z) = \frac{\Phi_{uu}(z)}{|D(z)|^2} = \frac{E\{u_n^2\}}{|D(z)|^2} \quad (24)$$

The mean square value is

$$E\{e_n^2\} = \frac{1}{2\pi j} \oint \Phi_{ee}(z) \frac{dz}{z} \\ = E\{u_n^2\} \cdot \frac{1}{2\pi j} \oint \frac{1}{|D(z)|^2} \frac{dz}{z} \quad (25)$$

OUTPUT ERROR TO SIGNAL RATIO

Quite often, one is interested in the error-to-signal ratio at the output. In terms of our notations, this quantity is $E\{e_n^2\}/E\{w_n^2\}$, or $E\{y_n'^2\}/E\{y_n^2\}$, which, by using Eqs. (14), (23), and (25), can be written as

$$\frac{E\{e_n^2\}}{E\{w_n^2\}} = \frac{1}{2\pi j} \oint \frac{1}{|D(z)|^2} \frac{dz}{z} \cdot \frac{\frac{1}{2\pi j} \oint \{ |B(z)|^2 + |A(z) \frac{N(z)}{D(z)}|^2 - 2C(z) \frac{N(z)}{D(z)} \} \Phi_{xx}(z) \frac{dz}{z}}{\frac{1}{2\pi j} \oint \frac{|N(z)|^2}{|D(z)|^2} \Phi_{xx}(z) \frac{dz}{z}} \quad (26)$$

UPPER BOUND FOR ROUND-OFF ERROR ACCUMULATION

From Eqs. (23) and (25), we may derive an upper bound of $E\{e_n^2\}$. Since

$$\begin{aligned} E\{u_n^2\} &\leq \frac{1}{2\pi j} \oint \Phi_{xx}(z) \frac{dz}{z} \\ &\quad \max_{|z|=1} [|B(z)|^2 + |A(z) \frac{N(z)}{D(z)}|^2 - C(z) \frac{N(z)}{D(z)} - C(1/z) \frac{N(1/z)}{D(1/z)}] \\ &= R_{xx}(0) \cdot \max_{|z|=1} [|B(z)|^2 + |A(z) \frac{N(z)}{D(z)}|^2 - C(z) \frac{N(z)}{D(z)} - C(1/z) \frac{N(1/z)}{D(1/z)}] \\ E\{e_n^2\} &\leq R_{xx}(0) \frac{1}{2\pi j} \oint \frac{dz}{z |D(z)|^2} \cdot \\ &\quad \max_{|z|=1} [|B(z)|^2 + |A(z) \frac{N(z)}{D(z)}|^2 - C(z) \frac{N(z)}{D(z)} - C(1/z) \frac{N(1/z)}{D(1/z)}] \end{aligned} \quad (27)$$

Similarly, an upper bound of the error-to-signal ratio at the output can be derived:

$$\begin{aligned} \frac{E\{e_n^2\}}{E\{w_n^2\}} &\leq \frac{1}{2\pi j} \oint \frac{dz}{z |D(z)|^2} \cdot \\ &\quad \max_{|z|=1} \{ \frac{|D(z)|^2}{|N(z)|^2} \cdot [|B(z)|^2 + |A(z) \frac{N(z)}{D(z)}|^2 - C(z) \frac{N(z)}{D(z)} - C(1/z) \frac{N(1/z)}{D(1/z)}] \} \end{aligned} \quad (28)$$

EXAMPLE

Here we take the second order filter considered in Sandberg's paper [2]:

$$w_n = x_n - (a_1 w_{n-1} + a_2 w_{n-2})$$

with $a_1 = -\sqrt{2}(1-.001)$ and $a_2 = (1-.001)^2$. Thus $N(z) = 1$, $D(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$, $|B(z)|^2 = q$, and $|A(z)|^2 = (a_1^2 + a_2^2) [(1+q)^3 - 1] + a_1 a_2 [(1+q)^2 - 1](z^{-1} + z)$, $C(z) = a_1 z^{-1} + a_2 z^{-2}$. Since $(1+q)^3 - 1 \approx 3q$ and $(1+q)^2 - 1 \approx 2q$, we have

$$\begin{aligned} & \left| \frac{D(z)}{N(z)} \right|^2 \cdot [|B(z)|^2 + |A(z) \frac{N(z)}{D(z)}|^2 - C(z) \frac{N(z)}{D(z)} - C(1/z) \frac{N(1/z)}{D(1/z)}] \\ &= q[1 + 2(a_1^2 + a_2^2) + a_1 a_2 (z^{-1} + z)] \end{aligned}$$

Its maximum on $|z|=1$ is seen easily to be approximately $(7+2\sqrt{2})q$. The integral $1/(2\pi j) \oint dz/(z|D(z)|^2)$ can be easily evaluated; its value is 0.5×10^3 . With $t=27$ the upper bound of $E\{e_n^2\}/E\{w_n^2\}$ as given by Eq. (28) is 3.3×10^{-7} . The bound given by Sandberg is 5.8×10^{-5} .

ACKNOWLEDGMENT

This research is sponsored by the Air Force Office of Scientific Research, Office of Aerospace Research, United States Air Force under AFOSR Grant 1333-67 and by the National Science Foundation under Grant GK-1439.

APPENDIX

STATISTICS OF $\theta_{n,k}$ AND $\phi_{n,k}$

The first and second moments of $\theta_{n,k}$ and $\phi_{n,k}$ as given by Eq. (5) can be evaluated straightforwardly. The result is tabulated below.

$$\begin{aligned} E\{\theta_{n,j}\} &= E\{\phi_{n,j}\} = 1 \quad \forall n, j \\ E\{\theta_{n,0}^2\} &= (1+q)^{M+2}, \quad E[\theta_{n,j}\theta_{n,k}] = (1+q)^{M+2-j}, \quad M \geq j > k \geq 0 \\ E\{\theta_{n,j}^2\} &= (1+q)^{M+3-j}, \quad j=1, \dots, M \\ \phi_{n,0} &= \phi_{n,0}^2 = 1 \quad E[\phi_{n,0}\phi_{n,j}] = 1 \\ E\{\phi_{n,1}^2\} &= (1+q)^{N+1} \\ E\{\phi_{n,j}^2\} &= (1+q)^{N+3-j}, \quad j=2, 3, \dots, N \\ E[\phi_{n,j}\phi_{n,k}] &= (1+q)^{N+2-j}, \quad N \geq j > k \geq 1 \\ E[\theta_{n,j}\phi_{n,k}] &= 1+q \quad k \neq 0 \\ &= 0 \quad k=0 \\ E[\theta_{n,j}\theta_{m,k}] &= E[\phi_{n,j}\phi_{m,k}] = E[\theta_{n,j}\phi_{m,k}] = 1, \quad n \neq m \\ q &= 1/3(2^{-2t}) \end{aligned}$$

REFERENCES

1. See, for example, J.B. Knowles and E. M. Olcayto, "Coefficient Accuracy in Digital Filter Response", IEEE Trans. Circuit Theory, Vol. CT-15, pp. 31-41, March, 1968.
2. I. W. Sandberg, "Floating-Point Round-off Accumulation in Digital Filter Realization", BSTJ, Vol. XLVI, pp. 1775-1791, October, 1967.
3. J. H. Wilkinson, Rounding Errors in Algebraic Processes, Prentice-Hall, Englewood Cliffs, New Jersey, 1963.

THE IDENTIFICATION OF LINEAR SAMPLED-DATA SYSTEMS BY DIRECT SEARCH MINIMIZATION

G. LEININGER AND H. R. MARTENS

Department of Electrical Engineering
State University of New York at Buffalo

ABSTRACT

A method is presented by which a discrete system model can be generated to represent a system of unknown characteristics. The modeling process utilizes the output time response corresponding to a deterministic input sequence. The procedure is based upon a direct search minimization which seeks to minimize a penalty function in the least squares sense. Consideration is given to the uniqueness of the discrete model in representing the output time response. Numerical results as obtained from test cases demonstrate the effectiveness of this modeling procedure.

DESCRIPTION OF PROCEDURE

The modeling technique requires a number sequence corresponding to the output time response of the discrete system to a specified deterministic input.

Expressing the output sequence c_i as a convolution of the system weighting sequence d_i and the z -transform of the input time function, $R(z)$, one obtains

$$[c_0 + c_1 z^{-1} + \dots] = [d_0 + d_1 z^{-1} + \dots] * R(z) \quad (1)$$

From this relation, the previously unknown system weighting sequence can be extracted as a function of the measured output sequence and the known input sequence. The system weighting sequence is then the defining system characterization.

This approach makes possible the use of many types of input time functions, such as kronecker delta, step, ramp, sinusoid, etc.

The simulation model is selected to have the form of a discrete transfer function, or equivalently a linear recursion equations, with unknown coefficients. Division of the denominator polynomial into the numerator polynomial yields the weighting sequence of the computer model for a given set of transfer function coefficients. The coefficients of the model are adjusted by minimizing a penalty function of the form

$$J = \sum_{i=0}^N \frac{(h_i - d_i)^2}{|d_i|} + \lambda (c_{ss} - e_{ss}) \quad (2)$$

where	h_i	represents the computer weighting sequence
	d_i	represents the system sequence to be modeled
	N	is the length of the output record
	e_{ss}	indicates the steady state value of the computer model
	c_{ss}	represents the desired steady state value
	λ	is a weighting constant

This penalty function has been selected to yield invariance with respect to stability and steady state gain.

The optimization procedure selected for the minimization of (2) is a direct search routine, adapted from Hooke and Jeeve's Pattern Search¹ and Mugele's Probing Technique².

In general, it must be assumed that the order of the system to be modeled is unknown. It is, however, under special conditions possible to have sufficient knowledge about the system to permit the specification of a system order. For situations where this is not the case, an automatic selection of the order of the transfer function has been incorporated into the computer program³.

The effect of the length of the output record and the presence of delay elements seriously effect the uniqueness of the digital model. The presence of delay elements not only is pertinent to the selection of the relative degree of the denominator polynomial but it fundamentally influences the choice of the initial order of the transfer function. It can be shown that if the order of the system (L) is less than the number of delays (p) then there is no solution. Furthermore, if $L \geq p$, then there are $L-p+1$ solutions for each L .

Table 1 shows the numerical data for a third order test case.

TABLE 1				
System Step Response		Extracted Weighting Sequence	Computed Weighting Sequence	Computed Step Response
0	1.0000	1.00000	1.00000	1.00000
1	1.6000	0.60000	0.60074	1.60074
2	0.9200	-0.6800	-0.67600	0.92473
3	0.6240	-0.29600	-0.29730	0.62742
4	0.6328	0.00880	0.00880	0.63623
5	0.5922	-0.04064	-0.4060	0.59562
6	0.5957	0.00359	0.00356	0.59919
7	0.5915	-0.00422	-0.00421	0.5949
8	0.5919	0.00043	0.00043	0.5954
⋮				
ss	0.5915			0.5925

REFERENCES

1. Hooke and Jeeves, "Direct Search Solution of Numerical and Statistical Problems" JACM Vol. 8, No. 2 (1960).
2. Mugele, R. "A Nonlinear Digital Optimizing Program for Process Control Systems" Spring Joint Computer Conference, pp. 15-32 (1962).
3. Leininger G. "The Determination of Discrete System Models by Direct Search Minimization" Master of Science Thesis, State University of New York at Buffalo (1968).

REPRESENTATION OF GENERAL LINEAR NETWORKS

R. SAEKS

Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

The technological advance of the recent past with its associated proliferation of new devices has necessitated a significant expansion in the scope of the various engineering theories. In the case of network theory, the once predominant RLC devices have been joined by various classes of distributed, time-variable and time-discrete network in addition to various mixed networks. In an effort to establish a theory of networks sufficiently general to cope with this expanded range of devices the author has in a number of previous papers presented a theory of "General Linear Networks", these being defined simply as a component operator on a Hilbert space together with a linear connection operator. Using such an approach highly general analysis and synthesis procedures have been obtained which are, however, applicable to the usual special cases upon replacement of the arbitrary component and connection operators by the appropriate differential and/or difference operators. The general approach thus allows the formulation of techniques applicable to "all" linear networks which may, however, be readily translated to the format of the various special cases which arise in engineering practice. Moreover, the simplicity of the general Hilbert space structure often eliminates "unnecessary" details of the special cases which might otherwise obscure the essence of the problem. In fact, the derivation of results in the general context is often easier than the corresponding derivation for the special case.

In this paper we consider the case of a general linear network, the components of which are characterized by a relation, aTb , on a Hilbert space rather than an operator. These include the various idealized non-solvable components such as operational amplifiers, nullators, and norators. If one assumes that aTb is linear it can, without loss of generality, be represented by a linear operator.

$$\pi : H^2 \rightarrow H^2 \quad (1)$$

such that

$$(a,b) \vdash^\pi 0 \quad (2)$$

if and only if (a,b) is an admissible pair of the general linear network.

Once the existence of π has been exhibited, it is employed in the formulation of synthesis procedures applicable to the entire class of linear networks. In this case a partial solution to the synthesis problem for a general linear network characterized by a passive scattering operator is known, hence we consider here only the problem of reducing the synthesis of a network characterized by a π operator to that of a passive network characterized by a scattering operator. This is achieved thru (the infinite dimensional analog of) negative resistor loading and results in the passive operator

$$S = \begin{bmatrix} 0 & 1/c \\ \pi P_1/c & \pi P_2/c^2 \end{bmatrix} \quad (3)$$

on H^4 which when loaded in negative resistors at its H^2 "ports" realizes π at its first H^2 ports. Here P_1 and P_2 are projections onto the first and second components respectively of H^2 and c is a sufficiently large positive real constant. When the relation, aTb , is itself passive one desires to realize it using passive components only, hence alternative synthesis by nullator loading is also formulated.

Finally we consider the process of applying the general theory to the various special cases of interest. This is most easily achieved thru the artifice of representing the π operator as a matrix of (real of complex) functions which when it exists, allows the π operator to be conveniently manipulated. In particular the existence of frequency and time domain representations for various commonly encountered function spaces (L_2, l_2, D_+) is considered.

The general theory presented in the paper thus allows the derivation of results in a highly general context which are, however, applicable to the various special case of interest after translation to the appropriate function space.

DERIVATION OF CANONICAL STATE EQUATIONS FOR A CLASS OF DISTRIBUTED SYSTEMS

J. A. RESH and I. C. GOKNAR
Michigan State University
East Lansing, Michigan

ABSTRACT

The paper extends a recent behavioral approach to state modelling to permit the generation of non-reduced state descriptions. The extension is then applied to derive canonical-form state equations, recently introduced without derivation, for a broad class of linear time-invariant distributed systems.

INTRODUCTION

The state of a system is conceived intuitively as an internal condition reflecting those aspects of the system's experience that affect its future behavior. The internal condition aspect of this conception is the basis for what might be called "the structural approach" to state modelling; an approach that calls for identification of energy or information storage mechanisms which support the internal condition. The experiential aspect has recently lead to an alternative "behavioral approach" to the problem, developed effectively by Zadeh [1, 2] with modifications by Resh [3]. For systems or components whose internal structures are insufficiently understood or too inaccessible to permit recognition of the memory-supporting mechanisms, the new approach is the only available alternative.

Until last year the behavioral approach offered little more than a comprehensive behavioral definition of state descriptions. The Allerton paper [4] added an important element: a general procedure for construction of reduced state descriptions from (complete) behavioral data. The present paper extends this procedure to permit the generation of non-reduced state descriptions.

The extension is then applied to a broad class of linear time-invariant distributed and mixed lumped-distributed systems. The resulting state descriptions were introduced at the Midwest Symposium last Spring [5], where discussion centered on their properties. The focal point here is their derivation. In a paper to be presented at the IEEE International Symposium in Miami this December [6], they are applied to distributed network synthesis.

REVIEW OF REDUCED STATE DESCRIPTIONS

Ref. [4] introduced the following procedure for constructing a reduced state description of an arbitrary causal system $\mathcal{R}$: First, partition the set of pre- t_0 admissible inputs into a family $\mathcal{H}_{t_0} = \{H_{t_0}[u_0] \mid u_0 \in U_{(-\infty, t_0]}\}$ of equivalence classes by the equivalence relation: u_0 is equivalent to u'_0 if, for each $u \in U_{(t_0, \infty)}$, the concatenation $u_0 u$ is an admissible input if and only if $u'_0 u$ is; and, if both are, then the (t_0, ∞) -portions of the responses to $u_0 u$ and $u'_0 u$ are the same. Second, choose for the state space $\Sigma(t_0)$ of $\mathcal{R}$ at time t_0 any set whose cardinality is the same as that of $\mathcal{H}_{t_0}$, and select any function $C_{t_0} : \Sigma(t_0) \xrightarrow{1-1} \mathcal{H}_{t_0}$ onto to establish the connection between

states and the system. Finally, define the input-output-state relation $\bar{A}_{(t_0, \infty)}$ as follows: Its domain consists of all state-input pairs (σ_0, u) with σ_0 in $\Sigma(t_0)$ and u in $U_{(t_0, \infty)}$ such that, for some u_0 in the equivalence class $C_{t_0}(\sigma_0)$, u_0 is an admissible input; and $\bar{A}_{(t_0, \infty)}$ is the (t_0, ∞) -portion of the response to u_0 .

A CONSTRUCTION FOR NON-REDUCED STATE DESCRIPTIONS

If the partition $\mathcal{H}_{t_0}$ in the foregoing development is replaced by a partition based on a sufficient condition for equivalence of input-histories rather than on a necessary and sufficient condition then the result will again be a state description of $\mathcal{R}$, but this time it need not be in reduced form.

Specifically, let the replacement for $\mathcal{H}_{t_0}$ be $\mathcal{H}'_{t_0} = \{H'_{t_0}[u_0] \mid u_0 \in U_{(-\infty, t_0]}\}$ where: (1) u_0 is in $H'_{t_0}[u_0]$ for each u_0 in $U_{(-\infty, t_0]}$; (2) if u_0 and u'_0 are both pre- t_0 inputs then either $H'_{t_0}[u_0] = H'_{t_0}[u'_0]$ or the two classes are disjoint; and (3) if u'_0 is in $H'_{t_0}[u_0]$ then u'_0 is equivalent to u_0 .

Verification of the state axioms (M1) and (S1) is straightforward. Axiom (M2) is satisfied by fiat. And verification of (S2) is easy once it is observed that the required state σ_1 is $C_{t_1}^{-1}(H'_{t_1}[u_0 u_1])$, where u_0 is any member of $C'_{t_0}(\sigma_0)$.

STATE EQUATIONS FOR DISTRIBUTED SYSTEMS

From now on we restrict our attention to convolutional objects, i. e. to systems for which $(u, y) \in \mathcal{R}$ is equivalent to $y = h * u$. Many such systems are describable, as discussed in Ref. [5], by state equations of the form:

$$\frac{dx_s(t)}{dt} = s \cdot x_s(t) + u(t) \quad (s \text{ in the complex numbers})$$

$$y(t) = \int_{\mathcal{L}^1} c(s) \cdot x_s(t) \cdot ds + \sum_{k=0}^K d_k \cdot u^{(k)}(t).$$

Although the generality of this representation is somewhat greater, the present proof is limited by the following assumptions:

1. The input space of the system consists of all real regular distributions based on locally bounded and measurable functions u such that, for some finite T dependent upon u , $u(t) = 0$ for all $t < T$;
2. $h = z + \sum_{k=0}^K d_k \cdot \delta(k)$, where K is a nonnegative integer or ∞ and z is a real regular distribution with $z(t) = 0$ for $t < 0$;
3. $z(t) \cdot \exp(-ct)$ is absolutely integrable over $(-\infty, \infty)$ for some finite c ; and
4. $Z(s)$ (defined as $\int_{-\infty}^{\infty} z(t) \cdot \exp(-st) \cdot dt$ for $\text{Re}[s] > c$, by Assumption 3) is summable along the path $c-i\infty$ to $c+i\infty$, i. e. $|\int_{-\infty}^{\infty} Z(c+i\omega) \cdot d\omega| < \infty$.

A number of observations are technically important and, for brevity, we merely list them:

1. As we are talking of regular distributions, all integrals are Lebesgue;

2. $z(t) \cdot l(t)$, a product of two distributions, is in this case well-defined since z is regular; and because z is causal by Assumption 2, we have $z = z(t) \cdot l(t)$;
3. From Assumptions 1 and 3 we know that z and all inputs are locally integrable;
4. Almost everywhere, $z(t) = (1/2\pi i) \int_{c-i\infty}^{c+i\infty} Z(s) \cdot \exp(st) \cdot ds$ [Ref. 7, p. 558];
5. From Assumption 4 it follows that $\int_{-\infty}^{\infty} |Z(c+i\omega)| d\omega < \infty$ [Ref. 8, p. 157];
6. The convolutions $h*u$, $z*u$ and $\delta^{(k)}*u$ are well defined in the distributional sense since all the distributions have supports bounded on the left [Ref. 9, p. 124, Th. 5.4.1]. Moreover, since both z and u are regular distributions, $(z*u)(t)$ is given by $\int_{-\infty}^{\infty} z(t-\tau) \cdot u(\tau) \cdot d\tau$ [Ref. 9, p. 126, Ex. 5.4.1].

We can prove now the

Lemma. $(z*u)(t) = (1/2\pi i) \int_{c-i\infty}^{c+i\infty} Z(s) \cdot [\exp(st) \cdot l(t)*u(t)] \cdot ds$.

Proof. $(z*u)(t) = (u*z \cdot l)(t)$
 $= u(t) * l(t) / (2\pi i) \int_{c-i\infty}^{c+i\infty} Z(s) \cdot \exp(st) \cdot ds$
 $= (1/2\pi) \int_{-T}^t u(\tau) \left[\int_{-\infty}^{\infty} Z(c+i\omega) \cdot \exp((c+i\omega)(t-\tau)) \cdot d\omega \right] d\tau$
 $= \exp(ct)/(2\pi) \int_{-T}^t u(\tau) \left[\int_{-\infty}^{\infty} Z(c+i\omega) \cdot \exp(-c\tau+i\omega(t-\tau)) \cdot d\omega \right] d\tau$.

But, since $u(\tau)$ and $\exp(-c\tau)$ are locally integrable and $Z(c+i\omega)$ is absolutely integrable,

$$\begin{aligned} \int_{-T}^t \left| \int_{-\infty}^{\infty} u(\tau) \cdot Z(c+i\omega) \cdot \exp(-c\tau+i\omega(t-\tau)) \cdot d\omega \cdot d\tau \right| \\ = \int_{-T}^t |u(\tau) \cdot \exp(-c\tau)| \left| \int_{-\infty}^{\infty} Z(c+i\omega) \cdot d\omega \right| d\tau \\ \leq M \int_{-T}^t |u(\tau) \cdot \exp(-c\tau)| d\tau \\ < \infty. \end{aligned}$$

Thus [Ref. 8, p. 206, Th. 280] we can interchange the order of integration to obtain:

$$\begin{aligned} (z*u)(t) &= \exp(ct)/(2\pi) \int_{c-i\infty}^{c+i\infty} Z(c+i\omega) \left[\int_{-T}^t u(\tau) \cdot \exp(-c\tau+i\omega(t-\tau)) \cdot d\tau \right] d\omega \\ &= (1/2\pi i) \int_{c-i\infty}^{c+i\infty} Z(s) \cdot [\exp(st) \cdot l(t)*u(t)] ds \quad \text{Q. E. D.} \end{aligned}$$

We're prepared now for the main issue. As we said in the section on general non-reduced state descriptions, the key to the search for such descriptions is a workable sufficient condition for equivalence of pre- t_0 inputs. The following theorem establishes such a condition for the present class of systems:

Theorem. u_1 and u_2 in $U_{(-\infty, t_0]}$ are equivalent if

$$\int_{-\infty}^{t_0} \exp(s(t_0 - \tau)) \cdot u_1(\tau) \cdot d\tau = \int_{-\infty}^{t_0} \exp(s(t_0 - \tau)) \cdot u_2(\tau) \cdot d\tau$$

for all complex s .

Proof. To be shown is that if u_1 and u_2 satisfy the conditions of the theorem then, for any u such that $(u_1 u, y_1)$ and $(u_2 u, y_2)$ are in $\mathcal{R}$, the responses $y_1(t)$ and $y_2(t)$ are equal for all $t > t_0$.

We have from the lemma that, for $m=1$ or $m=2$,

$$y_m(t) = (1/2\pi i) \int_{\mathcal{L}^-} Z(s) \int_{-\infty}^t \exp(s(t-\tau)) (u_m u)(\tau) \cdot d\tau \cdot ds + \sum_{k=0}^K d_k (u_m u)^{(k)}(t);$$

for $t > t_0$ this equation reduces to:

$$\begin{aligned} y_m(t) &= (1/2\pi i) \int_{\mathcal{L}^-} Z(s) \int_{-\infty}^{t_0} \exp(s(t-\tau)) \cdot u_m(\tau) \cdot d\tau \cdot ds \\ &\quad + (1/2\pi i) \int_{\mathcal{L}^-} Z(s) \int_{t_0}^t \exp(s(t-\tau)) \cdot u(\tau) \cdot d\tau \cdot ds + \sum_{k=0}^K d_k u^{(k)}(t) \\ &= (1/2\pi i) \int_{\mathcal{L}^-} Z(s) \cdot \exp(s(t-t_0)) \int_{-\infty}^{t_0} \exp(s(t_0-\tau)) \cdot u_m(\tau) \cdot d\tau \cdot ds \\ &\quad + (1/2\pi i) \int_{\mathcal{L}^-} Z(s) \int_{t_0}^t \exp(s(t-\tau)) \cdot u(\tau) \cdot d\tau \cdot ds + \sum_{k=0}^K d_k \cdot u^{(k)}(t); \end{aligned}$$

the theorem now follows easily, from this last expression for y_1 and y_2 .

As suggested by the theorem, we now define $H'_{t_0}[u_0]$, for each u_0 in $U_{(-\infty, t_0]}$, to consist of those functions u'_0 in $U_{(-\infty, t_0]}$ such that

$$\int_{-\infty}^{t_0} \exp(s(t_0-\tau)) \cdot u'_0(\tau) \cdot d\tau = \int_{-\infty}^{t_0} \exp(s(t_0-\tau)) \cdot u_0(\tau) \cdot d\tau$$

for each complex s . By the Theorem, this meets the criteria described in the section on non-reduced state descriptions.

Next, we need a convenient labeling set $\Sigma(t_0)$ for the partition $\mathcal{H}'_{t_0}$ just defined. We make the obvious choice: The label for $H'_{t_0}[u_0]$ will be the pattern of values $\{x_s(t_0) = \int_{-\infty}^{t_0} \exp(s(t_0-\tau)) \cdot u_0(\tau) \cdot d\tau \mid s \text{ complex}\}$ which characterizes its members. $\Sigma(t_0)$ itself is then the collection of all such patterns as u_0 ranges over all of $U_{(-\infty, t_0]}$.

Finally, we define the input-output-state relation $\bar{A}_{(t_0, \infty)}$ by evaluating the last equation in the proof of the theorem:

$$\bar{A}_{(t_0, \infty)}(\sigma_0, u)(t) = \int_{\mathcal{L}^-} \frac{Z(s)}{2\pi i} \cdot x_s(t) \cdot ds + \sum_{k=0}^K d_k \cdot u^{(k)}(t).$$

The development of the new state equations is completed by noting that $x_s(t)$ as defined above satisfies the differential state equation and that $\bar{A}_{(t_0, \infty)}(\sigma_0, u)(t)$ is just $y(t)$.

CONCLUSIONS

Ref.[5] can be accurately described as a presentation of the conclusions of the present paper.

The standard definition of equivalence of pre-to inputs has produced an unsatisfactory resolution of one minor aspect of the state description presented here: Components of the response which are

concentrated on t_0 are ignored because the definition concerns only the open interval (t_0, ∞) . Unless $K=0$, such components will arise from "mismatch interaction" between the pre- and post- t_0 inputs. It's not presently clear how to account for this with the generality provided by distribution theory.

REFERENCES

1. L. A. Zadeh and C. A. Desoer, Linear System Theory, McGraw-Hill, New York, 1963.
2. L. A. Zadeh, "The Concept of State in System Theory, " Views on General System Theory (ed. M. D. Mesarovic), Ch. 3, John Wiley and Sons, New York, 1964.
3. J. A. Resh, "Improvements in the State Axioms, " Conference Record, Tenth Midwest Symposium on Circuit Theory, Purdue University, Lafayette, Indiana, May, 1967.
4. J. A. Resh, "On the Construction of State Spaces, " Proceedings, Fifth Annual Allerton Conference on Circuit and System Theory, University of Illinois, Urbana, Illinois, October, 1967.
5. J. A. Resh, "On Canonical State Equations for Distributed Systems, " Conference Record, Eleventh Midwest Symposium on Circuit Theory, University of Notre Dame, Notre Dame, Indiana, May, 1968.
6. G. T. Daryanani and J. A. Resh, "Foster Distributed-Lumped Network Synthesis, " IEEE Transactions on Circuit Theory and the 1968 IEEE International Symposium on Circuit Theory, Miami Beach, Florida, December, 1968 (to appear).
7. S. Seshu and N. Balabanian, Linear Network Analysis, John Wiley and Sons, New York, 1959.
8. H. Kestelman, Modern Theories of Integration, Oxford at the Clarendon Press, 1937.
9. A. H. Zemanian, Distribution Theory and Transform Analysis, McGraw-Hill, New York, 1965.

G. O. MARTENS

Department of Electrical Engineering

University of Manitoba

Winnipeg, Manitoba, Canada

ABSTRACT

In this paper topological formulas are derived for the entries of a hybrid matrix binomial - which determines the A-matrix of RLC networks - and for the coefficients of the characteristic polynomial. These quantities are obtained by inspection of the network topology. Also, bounds for the natural frequencies, which do not require the formulation of the state equations, are derived and expressed in terms of the appropriate network elements. Finally, a numerical example is given to illustrate the method and the calculation of these bounds.

I. INTRODUCTION

The state equation formulation has been widely used in the analysis of electrical networks and systems in general [1]. These equations are expressed in a standard mathematical form which is particularly suitable for transient response calculations. On the other hand, the topological formulation for transfer functions has the advantage that the resulting formulas are stated directly in terms of the network elements which, together with the topology, describe the physical network [2]. Use of these formulas circumvents the necessity of evaluating determinants with polynomial entries and avoids numerical errors due to approximate cancellation of terms in the expansion.

In this paper the salient features of the two methods of network analysis are combined. Topological formulas are derived for a hybrid matrix binomial $H = p\Lambda + H$, which determines the A-matrix [3] ($A = -\Lambda^{-1}H$), as well as the coefficients of the characteristic polynomial of the network. The entries of Λ are obtained by inspection of the capacitive and inductive subnetworks, and those of H from the resistive subnetwork. Also, bounds on the natural frequencies, or eigenvalues, in terms of the network elements, are obtained by utilizing the theorems of Bendixson, Pick, Wittmeyer, Wegner and Gerschgorin [4].

The advantages of both formulations are thus realized: The entries of the hybrid matrix binomial H and the coefficients of the characteristic polynomial are obtained by inspection. Since these quantities are sums of element products, or ratios of such sums, the dependence on the individual network elements is clearly shown; also, numerical errors due to approximate cancellations are avoided. The bounds which are obtained provide a direct means of setting limits on the transient behavior by simply examining the values of the elements. Finally, an example is given to demonstrate the topological formulas and to illustrate the computation of the bounds.

II. DECOMPOSITION OF THE NETWORK INTO SUBNETWORKS

As is well known the A-matrix for RLC networks is obtained by choosing a normal tree (due to Bryant), partitioning the fundamental cut-set matrix and, then, eliminating all the nonstate variables. The state variables, of course, are the capacitive tree-branch voltages and the inductive-chord currents.

For the sake of completeness and to introduce the notation required in the following sections, the matrix state equation is briefly derived; the derivation is similar to that of Kuh and Rohrer [5]. Specifically, using Kirchhoff's Voltage and Current Laws, Ohm's Law as well as the orthogonality between the cut-set matrix Q and the circuit matrix B , a network without sources is described by the following matrix equation:

$$\begin{bmatrix} \frac{1}{p}S_1 & 0 & 0 & -Q_{CS}^T & 0 & 0 \\ 0 & R_2 & 0 & -Q_{CR}^T & -Q_{CR}^T & 0 \\ 0 & 0 & pL_3 & -Q_{CL}^T & -Q_{GL}^T & -Q_{rL}^T \\ Q_{CS} & Q_{CR} & Q_{CL} & pC_4 & 0 & 0 \\ 0 & Q_{GR} & Q_{GL} & 0 & G_5 & 0 \\ 0 & 0 & Q_{rL} & 0 & 0 & \frac{1}{p}r_6 \end{bmatrix} \begin{bmatrix} I_S \\ I_R \\ I_L \\ V_C \\ V_G \\ V_r \end{bmatrix} = 0 \quad (1)$$

Here S , R and L denote chord elastances, resistances and inductances, respectively; C , G and r denote tree-branch capacitances, conductances and reciprocal inductances, respectively; p is the differential operator denoting differentiation with respect to time; and Q , together with subscripts stands for the appropriate submatrices of the fundamental cut-set matrix partitioned according to element type.

The coefficient matrix of (1) will be called the primitive hybrid matrix of the network. If the nonstate variables I_S , I_R , V_G and V_r are eliminated from (1), the following matrix equation is obtained:

$$\begin{bmatrix} p\mathcal{L} + H_{11} & H_{12} \\ -H_{12}^T & pC + H_{22} \end{bmatrix} \begin{bmatrix} I_L \\ V_C \end{bmatrix} = 0 \quad (2)$$

where $\mathcal{L} = L_3 + B_{Lr}L_6B_{Lr}^T$, $L_6 = r_6^{-1}$, $C = C_4 + Q_{CS}C_1Q_{CS}^T$, $C_1 = S_1^{-1}$, $H_{11} = Q_{GL}^T G^{-1} Q_{GL}$, $G = G_5 + Q_{CR}G_2Q_{CR}^T$, $G_2 = R_2^{-1}$, $H_{22} = Q_{CR}R^{-1}Q_{CR}^T$, $R = R_2 + B_{RG}R_5B_{RG}^T$, $R_5 = G_5^{-1}$, and $H_{12} = -Q_{CL}^T + Q_{GL}^T G^{-1} Q_{GR}G_2Q_{CR}^T$.

It is noted that matrices $\mathcal{L}$ and C are symmetric and positive definite; H_{11} and H_{22} are symmetric and positive definite or positive semidefinite.

The coefficient matrix of (2) will be called the near-primitive hybrid matrix and it can be written in the form of a matrix binomial:

$$H = p\Lambda + H \quad (3)$$

where

$$\Lambda = \begin{bmatrix} \mathcal{L} & 0 \\ 0 & C \end{bmatrix} \quad H = \begin{bmatrix} H_{11} & H_{12} \\ -H_{12}^T & H_{22} \end{bmatrix}$$

If x denotes the state vector, then the network is described by

$$(p\Lambda + H)x = 0$$

or

$$p_x = -\Lambda^{-1} H_x \quad (4)$$

which is the standard state equation with the A-matrix equal to $-\Lambda^{-1} H$.

The above formulation leads to a natural decomposition of the network into a resistive, an inductive and a capacitive subnetwork. The resistive subnetwork N_R is obtained by replacing all tree-branch capacitances by voltage sources, all chord inductances by current sources and, in addition, open-circuiting all chord capacitances and short-circuiting all tree-branch inductances. It follows from (1) that N_R is characterized by the hybrid matrix H . The inductive subnetwork N_L is obtained by open-circuiting all noninductive chords and short-circuiting all noninductive tree-branches. Then, if a voltage source is connected in series with each chord inductance, it follows from (1) that N_L is characterized by

$$E_L = p \mathcal{L}_L I_L$$

where E_L is the vector of inserted voltage sources. Similarly, the capacitive subnetwork N_C is obtained by open-circuiting all noncapacitive chords and short-circuiting all noncapacitive tree-branches. Now, if a current source is connected in parallel with each tree-branch capacitance, N_C is characterized by

$$J_C = p C_V V_C$$

where J_C is the vector of current sources.

III. TOPOLOGICAL FORMULAS FOR THE ENTRIES OF H

The entries ℓ_{ij} of $\mathcal{L}$, and c_{ij} of C , are obtained from the well-known rules for writing down loop-impedance and node-pair-admittance matrices, respectively. For the sake of completeness these rules are now stated:

$$\ell_{ii} = \Sigma \text{ inductances in loop } i.$$

$$\ell_{ij} = \Sigma(+) \text{ inductances in both loops } i \text{ and } j, \text{ with a } + \text{ sign if the references of } i \text{ and } j \text{ agree; and with a } - \text{ sign otherwise.}$$

$$c_{ii} = \Sigma \text{ capacitances in cut-set } i.$$

$$c_{ij} = \Sigma(+) \text{ capacitances in both cut-sets } i \text{ and } j \text{ with } + \text{ sign if both references of } i \text{ and } j \text{ agree, or both disagree, with the capacitance reference; and with } - \text{ sign otherwise.}$$

For the purpose of obtaining topological formulas for the entries of H , the ports are numbered: 1, 2, ..., n_L , $n_L + 1$, ..., $n_L + n_C = n$ where n_L and n_C are the number of chord inductances and tree-branch capacitances, respectively. The port terminals corresponding to part i are labeled i and i' with the positive voltage reference at i and the current reference entering terminal i . Now, clearly, if the appropriate ports are open- and short-circuited, the topological formulas developed for two-port networks may be used. Therefore applying the formulas and notation of Seshu and Reed [2], the entries h_{ij} of H are given by the following formulas with the short-circuited ports listed after a solidus.

The symbol W denotes the sum of the 2-tree-admittance products and the subscripts indicate which terminals are in separate parts of the 2-tree; V denotes the sum of the tree-admittance products.

For: $i = 1, 2, \dots, n_L$

$$h_{ii} = \frac{W_{i,i'}}{V} \Big|_{n_L + 1, \dots, n} \quad (5)$$

$i, j = 1, 2, \dots, n_L$

$$h_{ij} = \frac{(W_{ij,i'j'} - W_{ij',i'j})}{V} \Big|_{n_L + 1, \dots, n} \quad (6)$$

$i = n_L + 1, \dots, n$

$$h_{ii} = \frac{V \Big|_{n_L + 1, \dots, i-1, i+1, \dots, n}}{V \Big|_{n_L + 1, \dots, n}} \quad (7)$$

$i, j = n_L + 1, \dots, n$

$$h_{ij} = \frac{(W_{ij',i'j} - W_{ij,i'j'})}{V \Big|_{n_L + 1, \dots, n}} \Big|_{n_L+1, \dots, i-1, i+1, \dots, j-1, j+1, \dots, n} \quad (8)$$

$i = 1, 2, \dots, n_L$ and $j = n_L + 1, \dots, n$

$$h_{ij} = \frac{(W_{ij,i'j'} - W_{ij',i'j})}{V \Big|_{n_L + 1, \dots, n}} \Big|_{n_L+1, \dots, j-1, j+1, \dots, n} \quad (9)$$

IV. TOPOLOGICAL FORMULAS FOR THE COEFFICIENTS OF THE CHARACTERISTIC POLYNOMIAL

The characteristic polynomial of the network is given by

$$\begin{aligned} d(p) &= \det(pI - A) \\ &= \frac{\det(p\Lambda + H)}{\det \Lambda} \end{aligned} \quad (10)$$

The polynomial $\det(p\Lambda + H)$ is related to the polynomial $\det Y_n$ - where Y_n is the node admittance matrix of the network - via the primitive hybrid matrix H_o :

$$\det H_o = \det \frac{1}{p} S_1 \det \frac{1}{p} r_6 \det R_2 \det G \det H$$

Alternately,

$$\det H_o = \det \frac{1}{p} S_1 \det R_2 \det pL_3 \det Y_p$$

where Y_p is the node-pair admittance matrix of the network. Therefore,

$$d(p) = \frac{\det Y_p \det pL_3 \det pL_6}{\det \Lambda \det G} \quad (11)$$

$$= \frac{p^{t_L} \det Y_n}{V_{N_L} V_{N_C} V_{N_R}} \Big|_{n_L + 1, \dots, n}$$

where t_L is the number of inductances in the network. Seshu and Reed [2] have shown that

$$\det Y_n = \frac{a_k p^{k+r} + \dots + a_{-r}}{p^r} \quad (12)$$

where

$$k = \begin{aligned} & \text{(number of capacitances)} \\ & - \text{(number of independent C-loops)} \\ & - \text{(number of independent L-cut-sets),} \end{aligned}$$

$$r = \begin{aligned} & \text{(number of inductances)} \\ & - \text{(number of independent L-loops)} \\ & - \text{(number of independent C-cut-sets)} \end{aligned}$$

and

$$a_k, a_{-r} \neq 0.$$

Since $d(p)$ is a monic polynomial,

$$a_k = V_{N_L} V_{N_C} V_{N_R} \Big|_{n_L + 1, \dots, n}$$

Thus the characteristic polynomial may be written in the form

$$d(p) = p^{t_L - r} (p^m + d_{m-1} p^{m-1} + \dots + d_0) \quad (13)$$

where

$$m = k + r$$

Therefore the formula for the j^{th} coefficient is

$$d_j = \frac{a_{j-r}}{a_k} = \frac{\sum \text{tree-element products}}{V_{N_L} V_{N_C} V_{N_R}} \Big|_{n_L + 1, \dots, n} \quad (14)$$

where the sum contains those trees T for which $j = (\text{number of capacitances in } T) - (\text{number of inductances in } T) + r$ and the parameters in the tree-element products are capacitance, conductance and reciprocal inductance.

At this point it may be observed that, if and only if, t_L equals r ,

$$\det H = \frac{a_{-r}}{a_k} \neq 0$$

that is, if and only if, there are no L-loops and C-cut-sets.

V. TOPOLOGICAL FORMULAS FOR THE BOUNDS ON THE NATURAL FREQUENCIES OF THE NETWORK

The natural frequencies, or eigenvalues, of the network are the zeros of the characteristic polynomial or, equivalently, of the determinant of the near-primitive hybrid matrix. To facilitate the derivations in this section, the following notation is used: the symmetric and skew-symmetric parts of H are denoted by H_s and H_k ; the smallest and largest eigenvalues of a generic matrix F are denoted by $\lambda_{\min}(F)$ and $\lambda_{\max}(F)$; the smallest and largest entries by $\min(F)$ and $\max(F)$; respectively; and the trace of F by $\text{tr}(F)$. Also, the special properties of Λ and H , listed after (2), are used.

If λ is an eigenvalue of H , then in general

$$\lambda = \alpha + j\beta$$

Pick's theorem [4] then gives the bound

$$|\beta| \leq \frac{|\max(H_k)|}{\lambda_{\min}(\Lambda)} \cot \frac{\pi}{2n} \quad (15)$$

Bendixson's theorem [4] states that α is bounded by the smallest and largest roots of

$$\det(p\Lambda + H_s) = \det(p\mathcal{L} + H_{11}) \det(pC + H_{22})$$

Then with the aid of the variational characterization of eigenvalues [4], Bendixson's bound for α yields

$$\min \left\{ \frac{\lambda_{\min}(H_{11})}{\lambda_{\max}(\mathcal{L})}, \frac{\lambda_{\min}(H_{22})}{\lambda_{\max}(C)} \right\} \leq -\alpha \leq \max \left\{ \frac{\lambda_{\max}(H_{11})}{\lambda_{\min}(\mathcal{L})}, \frac{\lambda_{\max}(H_{22})}{\lambda_{\min}(C)} \right\} \quad (16)$$

Lower and upper bounds for the smallest and largest eigenvalues, respectively, of the various matrices are now obtained:

$$\begin{aligned} \lambda_{\min}(\mathcal{L}) &= \min_{x \neq 0} \frac{x^T (L_3 + B_{Lr} L_6 B_{Lr}^T) x}{x^T x} \\ &\geq \min(L_3) \end{aligned} \quad (17)$$

$$\begin{aligned} \lambda_{\max}(\mathcal{L}) &= \max_{x \neq 0} \frac{x^T \mathcal{L} x}{x^T x} \\ &\leq \max(L_3) + \text{tr}(B_{Lr} L_6 B_{Lr}^T) \end{aligned} \quad (18)$$

Similarly

$$\lambda_{\min}(C) \geq \min(C_4) \quad (19)$$

$$\lambda_{\max}(C) \leq \max(C_4) + \text{tr}(Q_{CS} C_1 Q_{CS}^T) \quad (20)$$

Wittmeyer's and Wegner's theorem [4] is now applied to yield

$$\lambda_{\min}(H_{11}) \geq \det H_{11} \left(\frac{n_L - 1}{\text{tr}(H_{11})} \right)^{n_L - 1} \quad (21)$$

Murata [6] has given a formula for $\det H_{11}$ which may be obtained from (1):

$$\begin{aligned} \det H_{11} &= \frac{\begin{vmatrix} R & B_{RG}^R B_{LG}^T \\ B_{LG}^R B_{RG}^T & B_{LG}^R B_{LG}^T \end{vmatrix}}{\det R} \\ &= \frac{V|1, 2, \dots, n}{V|n_L + 1, \dots, n} \end{aligned} \quad (22)$$

Similarly,

$$\det H_{22} = \frac{V}{V|n_L + 1, \dots, n} \quad (23)$$

From (5),

$$\text{tr}(H_{11}) = \frac{\sum_{i=1}^{n_L} W_{i,i'}}{V} \left| \begin{matrix} n_L + 1, \dots, n \end{matrix} \right| \quad (24)$$

and from (7),

$$\text{tr}(H_{22}) = \frac{\sum_{i=n_L+1}^n V|n_L + 1, \dots, i-1, i+1, \dots, n}{V|n_L + 1, \dots, n} \quad (25)$$

$$\begin{aligned} \lambda_{\max}(H_{11}) &= \max_{x \neq 0} \frac{x^T H_{11} x}{x^T x} \\ &= \max_{x \neq 0} \frac{x^T (B_{LG}^R B_{LG}^T - B_{LG}^R B_{RG}^T R^{-1} B_{RG}^R B_{LG}^T) x}{x^T x} \\ &\leq \max_{x \neq 0} \frac{x^T (B_{LG}^R B_{LG}^T) x}{x^T x} \\ &\leq \text{tr}(B_{LG}^R B_{LG}^T) \end{aligned} \quad (26)$$

Similarly,

$$\lambda_{\max}(H_{22}) \leq \text{tr}(Q_{CR} G_2 Q_{CR}^T) \quad (27)$$

Now,

$$\text{tr}(B_{Lr}^L B_{Lr}^T) = \sum_{i=1}^{n_L} (\sum \text{inductances in loop } i \text{ of } N_L \text{ excluding the chord inductance}) \quad (28)$$

and

$$\text{tr}(Q_{CS}^C Q_{CS}^T) = \sum_{i=n_L+1}^n (\sum \text{capacitances in cut-set } i \text{ of } N_C \text{ excluding the tree-branch capacitance}) \quad (29)$$

Equations (17) to (29) lead to the definitions

$$m_1 = \frac{\frac{V|_{1,\dots,n}}{V|_{n_L+1,\dots,n}} \left[\frac{(n_L-1)V}{\sum_{i=1}^{n_L} W_{i,i'}} \middle|_{n_L+1,\dots,n} \right]^{n_L-1}}{\max(L_3) + \sum_{i=1}^{n_L} (\sum \text{ inductances in loop } i \text{ of } N_L \text{ excluding the chord inductance})} \quad (30)$$

$$m_2 = \frac{\frac{V}{V|_{n_L+1,\dots,n}} \left[\frac{(n_C-1)V|_{n_L+1,\dots,n}}{\sum_{i=n_L+1}^n V|_{n_L+1,\dots,i-1,i+1,\dots,n}} \right]^{n_C-1}}{\max(C_4) + \sum_{i=n_L+1}^n (\sum \text{ capacitances in cut-set } i \text{ of } N_C \text{ excluding the tree-branch capacitance})} \quad (31)$$

$$M_1 = \frac{\text{tr}(B_{LG} R_5 B_{LG}^T)}{\min(L_3)}$$

$$= \frac{\sum_{i=1}^{n_L} (\sum \text{ resistances in loop } i)}{\min(L_3)} \quad (32)$$

and

$$M_2 = \frac{\text{tr}(Q_{CR} G_2 Q_{CR}^T)}{\min(C_4)}$$

$$= \frac{\sum_{i=n_L+1}^n (\sum \text{ conductances in cut-set } i)}{\min(C_4)} \quad (33)$$

Using (17) to (33), the bound (16) for α yields

$$\min\{m_1, m_2\} \leq -\alpha \leq \max\{M_1, M_2\} \quad (34)$$

Note that when H_{11} , H_{22} or both are singular, then $\min\{m_1, m_2\} = 0$.

It can be shown that H_{11} is singular if, and only if, the subnetwork obtained by short-circuiting all capacitances contains L -loops (counting self-loops); and H_{22} is singular if, and only if, the subnetwork obtained by open-circuiting all inductances contains C -cut-sets.

The bound (15) for the imaginary part of the eigenvalue is rewritten as

$$|\beta| \leq \frac{\cot \frac{\pi}{2n}}{\min\{\min(C_4), \min(L_3)\}} \quad (35)$$

where use is made of the fact that the entries of H_k are voltage and current transfer ratios and, hence, not greater than one in magnitude.

A simplification arises when each chord inductance is connected in series with a resistance: $\lambda_{\min}(H_{11})$ is bounded below by the smallest of these resistances. Similarly, if each tree-branch capacitance is connected in parallel with a conductance, then $\lambda_{\min}(H_{22})$ is bounded below by the smallest of these conductances. Furthermore, if there are no other resistances or conductances, $\lambda_{\max}(H_{11})$ and $\lambda_{\max}(H_{22})$ are bounded above by the largest resistance and conductance, respectively.

For a network containing negative resistances or time-varying elements, which may be decomposed by Murata's method [6], H_{11} and H_{22} are not, a priori, positive definite or semidefinite and (34) is not applicable. Gerschgorin's theorem [4], however, may be applied. Then

$$\min_i (h_{ii} - \sum_{\substack{j \\ i \neq j}} |h_{ij}|) \leq \begin{cases} \lambda_{\min}(H_{11}) & i, j = 1, \dots, n_L \\ \lambda_{\min}(H_{22}) & i, j = n_L + 1, \dots, n \end{cases} \quad (36)$$

and

$$\max_i (h_{ii} + \sum_{\substack{j \\ i \neq j}} |h_{ij}|) \geq \begin{cases} \lambda_{\max}(H_{11}) & i, j = 1, \dots, n_L \\ \lambda_{\max}(H_{22}) & i, j = n_L + 1, \dots, n \end{cases} \quad (37)$$

These bounds together with (17)-(20) and (16) provide a bound for α which is a sufficient condition for asymptotic stability provided they are positive and the inductances and capacitances remain positive.

EXAMPLE 1:

Equations (5) to (9) and the rules of Section III are applied to the network of Fig. 1, as follows:

$$\mathcal{L} = \begin{bmatrix} L_1 + L_3 & -L_3 \\ -L_3 & L_2 + L_3 \end{bmatrix} \quad C = \begin{bmatrix} C_1 + C_2 & -C_4 & -C_4 \\ -C_4 & C_2 + C_4 & C_4 \\ -C_4 & C_4 & C_3 + C_4 \end{bmatrix} \quad H_{11} = \frac{1}{g_5 g_6} \begin{bmatrix} g_5 & 0 \\ 0 & g_6 \end{bmatrix}$$

$$H_{12} = \frac{g_5 g_6}{g_5 g_6} \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{bmatrix} \quad H_{22} = \frac{g_5 g_6}{g_5 g_6} \begin{bmatrix} g_1 + g_4 & -g_4 & -g_4 \\ -g_4 & g_2 + g_4 & g_4 \\ -g_4 & g_4 & g_3 + g_4 \end{bmatrix}$$

where a generic $g = 1/r$. Two sets of element values are considered: $L_1=1$, $L_2=2, L_3=2, C_1=2, C_2=1, C_3=2, C_4=3, g_1=1/2, g_2=1.5, g_3=1/3, g_4=1, -1, g_5=2$ and $g_6=1$. For the first set, (30)-(34) and the notes in Section V are applied. Then $0.03 \leq -\alpha \leq 4.84$, while by direct calculation $-\alpha_{\max}=0.64$ and $-\alpha_{\min}=0.21$. For the second set, (36), (37), (17)-(20) and (16) are applied. Here $0.083 \leq -\alpha \leq 6.0$ and by direct calculation $-\alpha_{\max}=0.55$ and $-\alpha_{\min}=0.14$.

VI. CONCLUSIONS

The topological formulas for the entries of the near-primitive hybrid matrix and the coefficients of the characteristic polynomial, relate these quantities directly to the symbolic network elements. Thus the matrix which characterizes the dynamic behavior of the network is obtained by inspecting the network topology.

In addition, bounds for the natural frequencies are obtained in terms of the symbolic elements. This provides a rapid means of establishing bounds on the transient behavior of the network by examining the element values.

Furthermore, a sufficient condition for asymptotic stability of a network with negative resistances or time-varying elements is given, subject to the conditions stated in Section V.

Finally, an example is used to demonstrate the topological formulas and the bounds on the natural frequencies.

ACKNOWLEDGMENT

The author wishes to thank K. Khosla for his assistance with the numerical calculations. This work was supported by the National Research Council of Canada, under Grant A-3813.

REFERENCES:

- [1] L.A. Zadeh and C.A. Desoer, Linear System Theory. New York: McGraw-Hill, 1963.
- [2] S. Seshu and M.B. Reed, Linear Graphs and Electrical Networks. Reading, Massachusetts: Addison-Wesley, 1961.
- [3] G.O. Martens, "On the Synthesis of RLC n-ports", Proc. Fifth Annual Allerton Conference on Circuit and System Theory, pp.739-747, Sept. 1967.
- [4] E. Bodewig, Matrix Calculus. New York: Interscience, 1956.
- [5] E.S. Kuh and R.A. Rohrer, "The State-Variable Approach to Network Analysis", Proc. IEEE, vol.53, pp.672-686, July 1965.
- [6] T. Murata, "Network Topology and Stability of Time-Varying RLC Networks", Trans. Circuit Theory, vol. CT-14, pp.306-313, Sept. 1967.

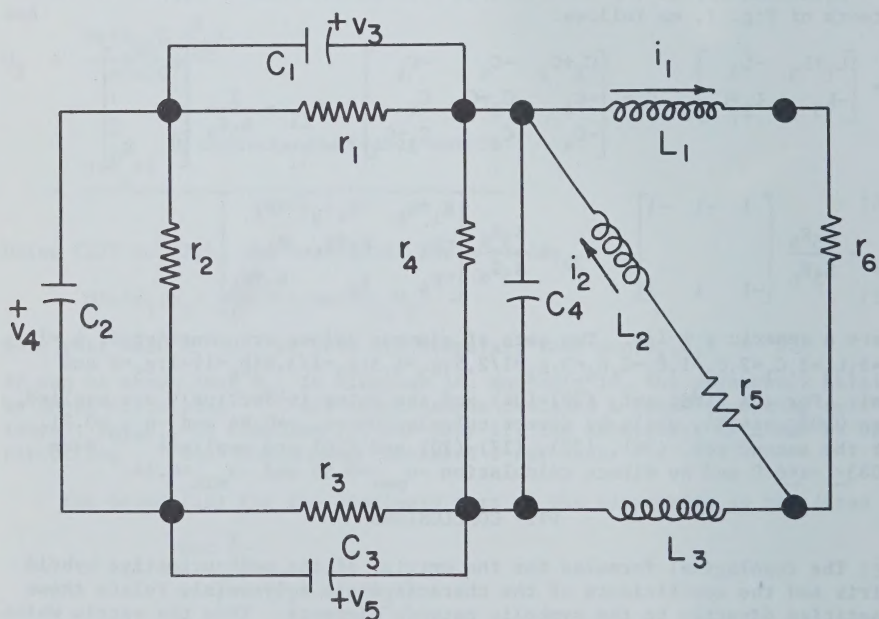


FIGURE 1: THE NETWORK FOR EXAMPLE 1.

THE SYNTHESIS OF A CLASS OF TRANSFORMERLESS n-PORT NETWORKS

DEAN J. SCHMIDLIN AND FRANK J. LUPO

New York University, Department of Electrical Engineering
Bronx, New York 10453

ABSTRACT

This paper introduces a special class of n-port networks called p-networks. A few select properties of p-networks and some prominent advantages of p-network synthesis are discussed. Finally, a method is presented for synthesizing an important class of p-networks from their short-circuit admittance parameter matrices.

I. INTRODUCTION

An outstanding problem in linear multiport synthesis is the realization of short-circuit admittance parameter matrices as passive RLC n-port networks without transformers. A number of well-known solutions exist for $n+1$ terminal^{1,3} and $2n$ terminal⁵ realizations, and a method² has been presented for synthesizing $(n+k)$ -terminal n-port networks on k-tree port-structures ($k=1,2,\dots,n$). This paper introduces another class of transformerless n-port networks called p-networks. These networks possess properties which make them useful in the realization of short-circuit admittance parameter matrices.

II. A SPECIAL CLASS OF n-PORT NETWORKS

The basic form of a special class of n-port networks called p-networks is illustrated in Figure 1. A p-network N is a passive transformerless RLC n-port which is composed of two n-port networks connected in series: 1) network N_0 , a transformerless $(n+1)$ -terminal n-port, and 2) network $\bar{N}$ a $2n$ -terminal n-port which consists of n disjoint impedances Z_i , each of which is associated with a single port. The composite network $\bar{N}$ (shown in Figure 1) consists therefore of a $(n+1)$ -terminal network N_0 to which is connected n series-impedances Z_i . Impedance Z_i is in series with port i of network N and is defined as the port series-impedance of port i . Some or all of these impedances may be zero. The number of nonzero port series-impedances is denoted by parameter p . If $p=0$ then network N is a transformerless $(n+1)$ -terminal n-port.

Let matrix $Y(s)$ be the short-circuit admittance parameter matrix of network N and let matrix $Y_0(s)$ be the short-circuit admittance parameter matrix of network N_0 . These two admittance matrices are related as follows,

$$Y_0 = [Y^{-1} - \bar{Z}]^{-1} \quad (1)$$

where

$$\bar{Z} = \text{Diag}[Z_1, Z_2, \dots, Z_n] \quad (2)$$

If $p=1$, then the relationship between the elements y_{ij} of matrix $Y(s)$ and the elements $y_{ij}^{(0)}$ of matrix $Y_0(s)$ is a simple one and is given below,

$$y_{ij}^{(0)} = y_{ij} + Z_m' y_{im} y_{mj} \quad (3)$$

$$y_{ij} = y_{ij}^{(0)} - Z_m'' y_{im}^{(0)} y_{mj}^{(0)} \quad (4)$$

where,

$$Z_m' = Z_m / (1 - Z_m y_{mm}) \quad (5)$$

$$Z_m'' = Z_m / (1 + Z_m y_{mm}^{(0)}) \quad (6)$$

and where Z_m is the only nonzero port series-impedance. If $p > 1$ then the relations between these two sets of short-circuit admittance parameters are more complex and are given in Reference 4. However, for any value of p , the short-circuit admittance parameters $y_{ij}^{(0)}$ may be calculated from the short-circuit admittance parameters y_{ij} by an iterative application of Equation (3). Similarly, the parameters y_{ij} may be determined from the parameters $y_{ij}^{(0)}$ by an iterative application of Equation (4). In this way, the p nonzero port series-impedances are extracted or inserted one at a time.

p -networks possess two important characteristics that make them useful in the realization of short-circuit admittance parameter matrices: (1) p -networks permit the use of nonideal voltage sources and, (2) the number of network branches in a p -network does not exceed $n(n+1)/2 + p$. In addition, p -network synthesis is capable of producing many equivalent networks from a single p -network realization.

In the realization of a given short-circuit admittance parameter matrix $Y(s)$ as a p -network, it is necessary to extract a set of series-impedances $Z_1, Z_2, \dots, Z_n$ so as to form a short-circuit admittance parameter matrix $Y_0(s)$ that is realizable as a transformerless $(n+1)$ -terminal n -port network. The matrix $Y_0(s)$ is then realized by the application of well known $(n+1)$ -terminal synthesis techniques.^{1,3} Finally, one realizes the matrix $Y(s)$ by simply reinserting the extracted impedances in the network N_0 that realizes $Y_0(s)$.

III. A PARTICULAR CLASS OF p -NETWORKS

Consider a resistive p -network N in which $p=n$ and $Z_i=R_0$ ($i=1,2,\dots,n$). Diagonal matrix Z in Equation (2) may be written as,

$$\bar{Z} = R_0 I \quad (7)$$

where Matrix I is the identity matrix. The short-circuit conductance parameter matrix G of network N is a real symmetric matrix and consequently may be decomposed into the following triple product,

$$G = Q \Lambda Q^t \quad (8)$$

where

$$\Lambda = \text{Diag}[\lambda_1, \lambda_2, \dots, \lambda_n] \quad (9)$$

is a diagonal matrix containing the eigenvalues of parameter matrix G and Q is the corresponding orthonormal eigenvector matrix. Since matrix Q is an orthogonal matrix,

$$QQ^t = I \quad (10)$$

substituting Equation (10) into Equation (7),

$$\bar{Z} = R_0 QQ^t = QR_0 I Q^t \quad (11)$$

taking the inverse of matrix G in Equation (8),

$$G^{-1} = Q \Lambda^{-1} Q^t \quad (12)$$

If matrix G_0 is the short-circuit conductance parameter matrix of the $(n+1)$ -terminal n -port network N_0 , then substitution of (11) and (12) into (1) and factoring results in,

$$G_0 = Q(\Lambda^{-1} - R_0 I)^{-1} Q^t \quad (13)$$

If

$$H = (\Lambda^{-1} - R_O I)^{-1} \quad (14)$$

Then Equation (13) may be written as,

$$G_O = Q H Q^t \quad (15)$$

The diagonal matrix H contains the eigenvalues of parameter matrix G_O . From Equations (8) and (15) it is seen that matrices G and G_O have the same orthonormal eigenvector matrix. If

$$H = \text{Diag}[\eta_1, \eta_2, \dots, \eta_n] \quad (16)$$

Then eigenvalue η_i ($i=1,2,\dots,n$) of parameter matrix G_O may be written from Equation (14) as,

$$\eta_i = \lambda_i / (1 - R_O \lambda_i) \quad (17)$$

If the parameter matrix G is singular then its inverse does not exist and at least one of its eigenvalues is zero. However, Equation (17) is valid even if $\lambda_i = 0$. If $\lambda_i = 0$, then $\eta_i = 0$ regardless of the value of R_O , and consequently the parameter matrix G_O is also singular. Since the parameter matrices G and G_O are positive semidefinite, the eigenvalues λ_i and η_i ($i=1,2,\dots,n$) are nonnegative. Applying this condition to eigenvalue η_i in Equation (17),

$$R_O \lambda_i < 1 \quad i=1,2,\dots,n \quad (18)$$

$R_O \lambda_i \neq 1$ because eigenvalue η_i must be finite. If λ_m is the maximum eigenvalue of parameter matrix G then an upper bound on port resistor R_O is,

$$R_O < 1/\lambda_m \quad (19)$$

Since $R_O \lambda_i < 1$ in Equation (17), eigenvalue η_i may be expanded into the following power series,

$$\eta_i = \sum_{m=0}^{\infty} \lambda_i^{m+1} R_O^m \quad (20)$$

The matrix H defined in Equation (14) may be written as,

$$H = \sum_{m=0}^{\infty} R_O^m \Lambda^{m+1} \quad (21)$$

substituting Equation (21) into Equation (15),

$$G_O = \sum_{m=0}^{\infty} R_O^m G^{m+1} \quad (22)$$

Equations (20) and (22) are useful in approximating the parameter matrix G_O or the eigenvalues η_i of matrix G_O .

This section introduced a special class of p-networks. In the next

section the synthesis of this class of networks will be considered.

IV. A SYNTHESIS PROCEDURE

This section presents a procedure for realizing a short-circuit conductance parameter matrix G as a resistive p-network having n identical port series-resistors. This synthesis technique is designed for machine computation which may be executed by means of the seven following operations:

- (1) Compute the eigenvalue matrix Λ and the corresponding orthonormal eigenvector matrix Q from parameter matrix G .
- (2) Choose a value for the normalized port resistor $\bar{R}_O = R_O \lambda_m$.
- (3) Compute a new eigenvalue matrix H using Equations (14) or (17).
- (4) Determine the short-circuit conductance parameter matrix G_O using Equation (15).
- (5) Check the realizability of parameter matrix G_O on $n + 1$ terminals.
- (6) If matrix G_O is realizable on $n + 1$ terminals then synthesize the corresponding $(n+1)$ -terminal n -port network N_O using known $(n+1)$ -terminal synthesis techniques. The parameter matrix G is then realized by inserting a resistance $R_O = \bar{R}_O / \lambda_m$ in series with each port of network N_O .
- (7) If matrix G_O is not realizable on $n + 1$ terminals then repeat steps (2) to (7) until such a matrix is found or until the normalized port resistor $\bar{R}_O$ is slightly less than unity.

From Equation (19) one sees that,

$$0 \leq \bar{R}_O < 1 \quad (23)$$

In many cases, it is sufficient to let the normalized port resistor $\bar{R}_O$ take on values from 0 to .95 in increments of .05. Currently, work is being done to obtain proper values of $\bar{R}_O$ more efficiently. If a realization of parameter matrix G is obtained for a particular value of $\bar{R}_O$, then varying $\bar{R}_O$ about this value produces a set of equivalent network realizations.

Example

Consider the following short-circuit conductance parameter matrix,

$$G = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 3.4 & 0.8 & 0.1 & -0.1 \\ 0.8 & 3.1 & 0.9 & 0.4 \\ 0.1 & 0.9 & 3.2 & 0.9 \\ -0.1 & 0.4 & 0.9 & 3.2 \end{bmatrix} \end{matrix} \quad (24)$$

where the units of the short-circuit conductance parameters g_{ij} in matrix (24) are in millimhos. Matrix (24) is unrealizable as a transformerless 5-terminal 4-port network with nonnegative resistors. Consequently, the normalized port resistor $\bar{R}_O$ must be greater than zero. Letting $\bar{R}_O = .6$ results in the following short-circuit conductance parameter matrix G_O ,

$$G_O = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 6.347 & 2.481 & 0.771 & 0.052 \\ 2.481 & 6.144 & 3.011 & 1.686 \\ 0.771 & 3.011 & 6.450 & 2.935 \\ 0.052 & 1.686 & 2.935 & 6.010 \end{bmatrix} \end{matrix} \quad (25)$$

Matrix (25) is a uniformly tapered matrix and consequently, it is realizable on the linear port-tree shown in Figure 2. The values of the branch conductances of the $(n+1)$ -terminal n -port network N_O are given by the upper triangular portion of the matrix,¹

$$\bar{G}_O = T G_O T \quad (26)$$

where,

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad (27)$$

element $\bar{g}_{ij}^{(o)}$ ($j \geq i$) in matrix $\bar{G}_O$ is the conductance that is connected between the node of negative polarity of port i and the node of positive polarity of port j . Applying transformation (26) to matrix (25),

$$\bar{G}_O = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 3.866 & 1.710 & 0.719 & 0.052 \\ & 1.423 & 0.606 & 1.634 \\ & & 2.190 & 1.249 \\ & & & 3.075 \end{bmatrix} \end{matrix} \quad (28)$$

where the lower triangular portion of matrix (28) has been omitted for convenience. Since the maximum eigenvalue λ_m of matrix (24) is 4.813×10^{-3} , the value of the port resistor $R_O = \bar{R}_O / \lambda_m$ is 124.66 ohms. The realization of short-circuit conductance parameter matrix G is shown in Figure 3.

Equivalent network realizations may be obtained by studying the behavior of the elements in the upper triangular portion of branch conductance value matrix $\bar{G}_O$ as a function of the normalized port resistance $\bar{R}_O$. A realization is obtained at a particular value of $\bar{R}_O$ if all the elements in the upper triangular portion of $\bar{G}_O$ are nonnegative at that value. Applying transformation (26) to matrix (24),

$$T G T = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 2.6 & 0.7 & 0.2 & -0.1 \\ & 1.5 & 0.3 & 0.5 \\ & & 1.8 & 0.5 \\ & & & 2.3 \end{bmatrix} \end{matrix} \quad (29)$$

At $\bar{R}_O = 0$, matrix $\bar{G}_O$ is equal to matrix (29). Figure 4 shows the variation of elements $\bar{g}_{14}^{(o)}$ and $\bar{g}_{44}^{(o)}$ with $\bar{R}_O$. $\bar{g}_{14}^{(o)}$ is the only element in the upper portion of $\bar{G}_O$ that is negative at $\bar{R}_O = 0$, and $\bar{g}_{44}^{(o)}$ is the first initially positive element in the upper portion of $\bar{G}_O$ to become negative. From the graph in Figure 4, one sees that a realization of matrix (24) on the port-structure shown in Figure 3 exists for values of $\bar{R}_O$ within the following range,

$$.56 \leq \bar{R}_O \leq .94 \quad (30)$$

The above class of p-networks is important because it permits the use of n nonideal voltage sources whose internal resistances may be as large as R_0 . For the example presented in this section a network realization exists in which the internal resistance of any source may be as large as 196 ohms.

V. CONCLUSION

This paper has introduced a special class of transformerless n -port networks called p-networks. In addition, a synthesis procedure has been presented for realizing a short-circuit conductance parameter matrix G as a resistive p-network having n identical port series-resistors. Some important advantages of p-network synthesis are: (1) p-networks permit the use of nonideal voltage sources, (2) the number of network branches in a p-network does not exceed $n(n+1)/2 + p$ and, (3) p-network synthesis is capable of producing many equivalent networks from a single p-network realization.

Presently, attention is being given to the problem of: (1) extending the synthesis procedure presented in section 4 to include frequency sensitive p-networks, and (2) realizing short-circuit admittance parameter matrices as p-networks having arbitrary port series-impedances.

ACKNOWLEDGMENT

The authors are indebted to the National Science Foundation for their partial support of this work (NSF Grant GK-1044).

REFERENCES

1. E.A. Guillemin, Theory of Linear Physical Systems, John Wiley and Sons, Inc., New York, New York, May 1963.
2. F.J. Lupo, "The Synthesis of Transformerless n -Port Networks on Multi-tree Port-Structures," IEEE Transactions on Circuit Theory, Vol. CT-15, September 1968.
3. D.J. Schmidlin and F.J. Lupo, "The Determination of Port Orientation Matrices From Short-Circuit Admittance Parameter Matrices," New York University Technical Report 400-172, November 1967.
4. D.J. Schmidlin and F.J. Lupo, "Transformerless n -Port Networks With Nonzero Port Immittances," New York University Technical Report (in preparation).
5. P. Slepian and L. Weinberg, "Synthesis Applications of Paramount and Dominant Matrices," Proc. National Electronics Conference, Vol. 14, pp. 611-631, 1958.

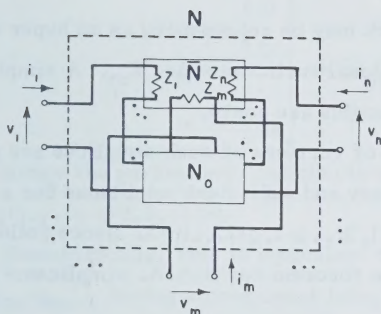


FIG. 1
THE BASIC FORM
OF A p - NETWORK

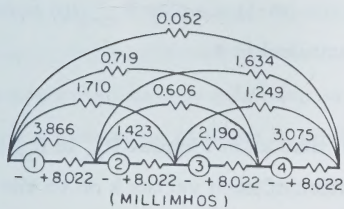


FIG. 3
REALIZATION OF
MATRIX (24)



FIG. 2
A LINEAR PORT-TREE

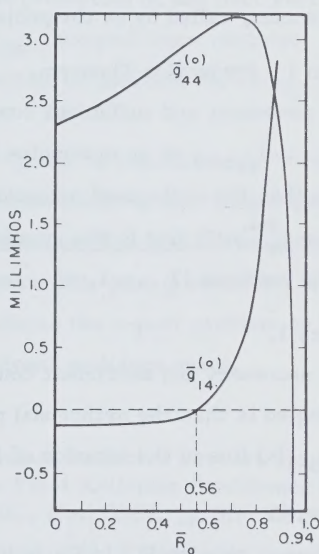


FIG. 4
VARIATION OF
 $\bar{g}_{14}^{(o)}$ AND $\bar{g}_{44}^{(o)}$
WITH $\bar{R}_0$

A GEOMETRIC INTERPRETATION OF THE PARAMOUNT CONDITION OF RESISTIVE N-PORTS*

D.J.H. MOORE
University of Newcastle
Australia

and G.E. SHARPE
The City College of New York
and Columbia University

ABSTRACT

An $(m+1)$ -terminal resistive network may be represented as an hyper acute angled (H.A.A.) simplex P_m in m -dimensional euclidean space E_m . A simplex is H.A.A. if all its first and higher order angles are acute.

Several lemmas on the distribution of vertices of such simplices are given and in particular we establish the necessary and sufficient conditions for a vertex q to be inside the simplex with vertices $1, 2, \dots, q-1, q+1, \dots, m+1$. Hence follows the main theorem, called by us the projection theorem for H.A.A. simplices:-

Theorem 1 - Projection Theorem.

A necessary and sufficient condition for the dihedral angles $\theta_{p1}, \theta_{p2}, \dots, \theta_{p, m+1}$ of an m simplex P_m , with vertices $1, 2, \dots, m+1$ in E_m , to be acute is that the orthogonal projection of vertex p onto the hyperplane $\pi(1, 2, \dots, p-1, p+1, \dots, m+1)$ lies in the interior of the $(m-1)$ simplex $P_{m-1}(p)$ formed from the vertices $1, 2, \dots, p-1, p+1, \dots, m+1$ contained in π .

Corollary 1.

A necessary and sufficient condition that an m simplex P_m in E_m be hyper-acute angled is that the orthogonal projection of each vertex k on to the $(m-1)$ face $P_{m-1}(k)$ lies in the interior of $P_{m-1}(k)$.

Corollary 2.

If an m simplex P_m in E_m is hyperacute angled then the orthogonal projection of any vertex of P_m on to any r face P_r ($r < m$) of P_m lies in the interior of P_r .

The convexity of any m simplex P_m ensures also:-

Corollary 3.

If x is any point in the interior of a hyperacute angled simplex P_m in E_m , then the orthogonal projection of x on to any r face P_r ($r < m$) of P_m lies in the interior of P_r .

* This research was supported in part by the National Science Foundation under Grant NSF GK-719.

We apply the projection theorem to a geometrical discussion of the paramount condition of resistive n-ports.

In particular, the O.C. $n \times n$ impedance matrix $\underline{Z}$ of an n-port inscribed on $2n$ terminals $1, 2, \dots, 2n$ defined by position vectors $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_{2n}$, may be expressed by $\underline{Z} = \underline{A} \underline{A}^T$, where

$$\underline{A} = \begin{bmatrix} \underline{a}_1 \\ \underline{a}_2 \\ \vdots \\ \underline{a}_n \end{bmatrix} = [a_{pq}]$$

Hence the paramount condition is equivalent to the requirement $a_{qq} \geq |a_{pq}|$ ($p=q, q+1 \dots n; q=1, 2, \dots, n$).

Geometrically, this is equivalent to the projection of any port vector, say $\underline{a}_n = \underline{x}_{2n} - \underline{x}_{2n-1}$ having a component lying in P_{2n-4} formed from vertices $1, 2, \dots, 2n-3$, which is smaller than the projection of position vector $\underline{x}_{2n-2}$ onto P_{2n-4} .

Because a resistive n-port ($n \leq m \leq 2n-1$) may be inscribed on $(m+1)$ -terminals (vertices), the projection theorem allows this simple geometric interpretation of the paramount condition. Because this condition is known to be necessary but insufficient for the realization of n-port $\underline{Z}$ matrices, the approach presented here is valuable, as it reduces the n-port problem to the solution of a geometric problem in m-dimensional euclidean space.

REFERENCES

1. Sharpe, G. E., "Solution of the $(m+1)$ -Terminal Resistive Network Problem by Means of Metric Geometry", Proc. of the First Asilomar Conference on Circuits and Systems, Pacific Grove, Calif., Nov. 1967, pp. 319-328.
2. Sharpe, G. E., "A Theorem on Resistive Networks", Electronics Letters, Vol. 3, No. 10, October 1967, pp. 444-445.
3. Moore, D. J. H., "A Geometric Theory for Networks with Positive Element Values", Proc. of the Hawaii International Symposium on System Science, Honolulu, Hawaii, Jan. 1968, pp. 246-250.
4. Moore, D. J. H. & Sharpe, G. E., "Metric Transformation of an $m+1$ terminal resistive Network into a Hyper-Acute-Angled Simplex p_m in Euclidean Space E_m ", Proc. of the Eleventh Mid-West Symposium on Circuit Theory, Univ. of Notre Dame, May 1968, pp. 184-191.
5. Sharpe, G. E. and D. J. H. Moore, "Transfer Impedances and the No-Amplification Property of Resistive Networks", Proc. IEEE, Vol. 56, No. 6, June 1968, pp. 1116-1117.

CLASS INVARIANT GENERATION OF POSITIVE REAL FUNCTIONS

N. K. BOSE

Department of Electrical Engineering
University of Pittsburgh
Pittsburgh, Pennsylvania 15213

ABSTRACT

Generation of positive real functions by elementary operations is a subject of considerable interest in network realizability theory. It is known, for example, that successive simultaneous differentiation of the numerator and denominator polynomials of a RC, RL, or LC driving-point function, results in functions, which are themselves, respectively the driving point functions of RC, RL, or LC networks. It is noticed that these generated functions belong to the same "class" of network functions as the original one. In this context, two p.r. functions are said to be class invariant, if and only if, they are both realizable by the same type of network elements. Some other known generating procedures do not satisfy this class invariance criterion.

In this paper, attention is directed toward new methods of systematic generation of class invariant positive real functions. Use is made of Kronecker sums and products of matrices, and the companion matrices associated with polynomials. It is shown that Kronecker's sum and product of matrices can be used to generate a very wide class of RC(RL) positive real functions, and Kronecker's sum of matrices utilized to generate class invariant LC p.r.f.'s. The generated p.r.f.'s are found to have interesting relationships with the original p.r.f.

Other methods of systematic generation are also studied. Of special interest is the case, when for certain functions, replacement of coefficients by their complementary values (i.e. a_r by a_{n-r} in a polynomial of degree n) maintains invariant of the positive real property but destroys the class invariant property. Possible problems for future research in this area are mentioned.

INTRODUCTION

Considerable amount of literature¹⁻⁷ is available on the generation of positive real functions. These generation procedures not only supply important properties of p.r.f.'s, but in some cases have resulted in new synthesis techniques⁸ and new coefficient relations¹. In this paper, new methods of systematic generation are presented.

The following definitions are quoted for convenience⁹.

Definition 1. Let A be an m-dimensional matrix and B an n-dimensional matrix. The mn-dimensional matrix A X B is called the Kronecker product of A and B and written

$$A \times B = (a_{ij} B)$$

Definition 2. The characteristic roots of A X B are $\lambda_i y_j$, where λ_i ($i=1,2,\dots,m$) are the characteristic roots of A and y_j are the characteristic roots of B.

Definition 3. The characteristic roots of the Kronecker sum of A and B, given by $C = I_m \times B + A \times I_n$ are $\lambda_i + y_j$ ($i=1,2,\dots,m$, $j=1,2,\dots,n$), where I_m and I_n are identity matrices of order m and n, respectively.

Definition 4. The companion matrix D, associated with a polynomial, $P(s) = s^n + a_1 s^{n-1} + \dots + a_n$, is given by

$$D = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & & 1 & 0 \end{bmatrix}$$

and then, $|D - SI| = P(s)$.

Definition 5. Kronecker powers are special cases of Kronecker products. In particular,

$$A^{(2)} = A \times A, \quad A^{(k+1)} = A \times A^{(k)}$$

are Kronecker powers of degree 2 and k+1, respectively. The alternation of poles and zeros of two-element kind dissipative networks, along with the foregoing definitions, enable one to arrive at the following assertions at once.

Assertion 1. If $Z_1(s) = \frac{P_1(s)}{Q_1(s)} = \frac{s^n + a_1 s^{n-1} + \dots + a_n}{s^n + b_1 s^{n-1} + \dots + b_n}$ is a RC(RL)

driving point function and if D_1 and D_2 are the companion matrices associated with polynomials $P_1(s)$ and $P_2(s)$, respectively, then,

$$Z_k(s) = \frac{|D_1^{(k)} - SI|}{|D_2^{(k)} - SI|}, \text{ for } k = 2, 3, \dots,$$

are also RC(RL) driving-point functions. The open-circuit and short-circuit natural frequencies of each of the derived functions $Z_k(s)$ are either integer powers or mutual products (taken two at a time) of the open-circuit and short-circuit natural frequencies of the original function $Z_1(s)$.

Assertion 2. If

$$Z_1(s) = \frac{P_1(s)}{Q_1(s)} = \frac{s^n + a_1 s^{n-1} + \dots + a_n}{s^n + b_1 s^{n-1} + \dots + b_n}$$

is a RC(RL) driving-point function, and if D_1 and D_2 are the companion matrices associated with polynomials $P_1(s)$ and $P_2(s)$, respectively, then

$$Z(s) = \frac{|(I_n \times D_1 + D_1 \times I_n) - s I_{2n}|}{|(I_n \times D_2 + D_2 \times I_n) - s I_{2n}|},$$

is also a RC(RL) driving-point function. The open-circuit and short-circuit natural frequencies of $Z(s)$ are mutual sums (taken two at a time) of the open and short-circuit natural frequencies, respectively, of $Z_1(s)$.

Assertion 3. If

$$Z_1(s) = \frac{P_1(s)}{Q_1(s)}$$

is a LC driving-point function, and D_1 and D_2 are defined as in the assertion above, then,

$$Z(s) = \frac{|(I_n \times D_1 + D_1 \times I_n) - s I_{2n}|}{|(I_n \times D_2 + D_2 \times I_n) - s I_{2n}|}$$

is also a LC driving-point function. In this case also, conclusions similar to that arrived in assertion 2, for the natural frequencies, are valid.

It is also clear, that for functions representing RLC canonic forms, whose poles and zeros alternate on a straight line symmetrically located with respect to the real axis in the left-half plane or its boundary¹⁰, the Kronecker sum can again be used to generate p.r.f.'s as in assertions 2 and 3. Generation, using Kronecker products is, generally invalid, in these cases.

It is noticed that the generated functions in assertions 1, 2, and 3 are class invariant.

GENERATION OF POSITIVE-REAL FUNCTIONS USING TRANSFORMATION OF EQUATIONS

It is possible, in many instances, without knowing the values of the roots of an equation in terms of the coefficients, to transform it by elementary substitutions, or by the aid of the symmetric functions of the

roots, into another equation whose roots shall have certain assigned relation to the roots of the proposed¹¹. Two well known transformations, to be used in this paper are quoted below as definitions¹².

Definition 6. To transform an equation,

$$P(s) = 0 = a_0 s^n + a_1 s^{n-1} + \dots + a_n = a_0 (s - \alpha_1)(s - \alpha_2) \dots (s - \alpha_n),$$

into another whose roots are m times that of the first, put $y = ms$, and substitute $\frac{y}{m}$ for s in the identity and multiply by m^n to get,

$$a_0 y^n + m a_1 y^{n-1} + \dots + m^n a_n = a_0 (y - m\alpha_1)(y - m\alpha_2) \dots (y - m\alpha_n),$$

Definition 7. To transform an equation,

$$P(s) = 0 = a_0 s^n + a_1 s^{n-1} + \dots + a_n = a_0 (s - \alpha_1)(s - \alpha_2) \dots (s - \alpha_n),$$

into one whose roots are the reciprocals of the roots of the first equation, put $s = \frac{1}{y}$, and multiply by y^n , to get

$$a_n y^n + a_{n-1} y^{n-1} + \dots + a_0 = a_n (y - \frac{1}{\alpha_1})(y - \frac{1}{\alpha_2}) \dots (y - \frac{1}{\alpha_n}) = 0$$

Definition 6 leads to the following assertion.

Assertion 4. If

$$Z(s) = \frac{P(s)}{Q(s)} = \frac{a_0 s^n + a_1 s^{n-1} + \dots + a_n}{b_0 s^m + b_1 s^{m-1} + \dots + b_m}$$

is the driving-point function of any two-element kind RC, RL, or LC network (of course, some of the coefficients in $Z(s)$, will vanish, depending upon the element type involved), then,

$$Z_k(s) = \frac{P(s)}{Q(s)} = \frac{a_0 s^n + k a_1 s^{n-1} + \dots + k^n a_n}{b_0 s^m + k b_1 s^{m-1} + \dots + k^m b_m},$$

where k is any real positive constant, is also a driving-point function respectively, representing, a RC, RL, or LC network.

It is noted that in this case also, the generated function is class invariant.

Definition 7 leads to the following assertion.

Assertion 5. If

$$Z(s) = \frac{P(s)}{Q(s)} = \frac{a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n}{b_0 s^n + b_1 s^{n-1} + \dots + b_{n-1} s + b_n},$$

is the driving-point function of a RC(RL) network, then,

$$Z_1(s) = \frac{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}{b_n s^n + b_{n-1} s^{n-1} + \dots + b_1 s + b_0}$$

is the driving-point function of a RL(RC) network.

This result is consistent with the coefficient inequalities¹,

$$\frac{a_0}{b_0} < \frac{a_1}{b_1} < \frac{a_2}{b_2} \dots < \frac{a_n}{b_n},$$

for $Z(s)$, a RC driving-point impedance or a RL driving-point admittance function, and $Z_1(s)$, a RL driving-point impedance function or a RC driving-point admittance function. In this case, it is readily seen, that the class invariant property is destroyed though the positive real property, is, as before, invariant. Furthermore, it is obvious, that if $Z(s) = \frac{P(s)}{Q(s)}$ is an LC driving-point function, then $Z_1(s)$, obtained as in assertion 5, after replacing the numerator and denominator coefficients, respectively, by their complementary values, is also a LC driving-point function, thus satisfying the class-invariant criterion. Whether, replacement of the coefficients by their complementary values will maintain the positive real property invariant, when the original function is any given positive real function, is the next question. That, if is true for the biquadratic,

$$Z(s) = \frac{a_0 s^2 + a_1 s + a_2}{b_0 s^2 + b_1 s + b_2},$$

follows at once from the fact that if $Z(s)$ is p.r.,

then,

$$Z_1(s) = \frac{a_2 s^2 + a_1 s + a_0}{b_2 s^2 + b_1 s + b_0},$$

is also p.r. That it is true for any positive real function follows from the well known fact that if $Z(s)$ is p.r., then $Z(\frac{1}{s})$ is p.r. Note that the zeros and poles of $Z(\frac{1}{s})$ are the reciprocal of the zeros and poles of $Z(s)$. This observation leads to the next assertion.

Assertion 6. If $Z(s) = \frac{P(s)}{Q(s)}$ is any p.r.f., then replacement of the coefficients of the numerator and denominator polynomials by their complementary values, leaves the p.r. property invariant.

RECOMMENDATIONS

It is expected that the theory of transformations, symmetric functions, covariants and invariants will yield more useful results than is available to date regarding the enpressing of the necessary and

sufficient conditions for positive reality solely in terms of the coefficients of a real rational function. Effertz's work¹³ in this direction though academically stimulating, leaves scope for improvement, when feasibility of practical implementation is the criterion. Extension of the work to positive real matrices is likely to be of interest.

REFERENCES

1. N. K. Bose and F. M. Reza, "Derived Positive Real Functions and Matrices and Consequent Coefficient Relations", Fifth Annual Allerton Conference on Circuit and System Theory, pp. 461-469, 1967.
2. F. M. Reza, "On the Generation of Positive Real Functions", IEEE Trans. on Ckt. Theory, Vol. CT-11, p. 431, September 1964.
3. F. M. Reza, "A Multiplication Theorem for Positive Real Functions", Proc. Am. Math. Soc., Vol. 9, No. 3, pp. 496-499, June 1958.
4. R. J. Duffin, "Elementary Operations which Generate Network Matrices", Amer. Math. Soc. Proc. 1, Vol. 6, No. 3, pp. 335-339, 1955.
5. S. Seshu and N. Balabanian, "Transformations of Positive Real Functions", Trans. IRE, CT-4, December 1957.
6. N. K. Bose, "Positive Real Function and Lyapunov Function Generation", to be published in IEEE Trans. PGAC, December 1968.
7. P. I. Richards, "A Special Class of Functions with Positive Real Part in a Half-Plane", Duke Math. J. 14, 1947.
8. R. Bott and R. J. Duffin, "Impedance Synthesis Without Use of Transformers", J. Appl. Phys., 20, 816, 1949.
9. R. Bellman, "Introduction to Matrix Analysis", McGraw-Hill Book Co. Inc., 1960.
10. F. M. Reza, "RLC Canonic Forms", Journal of Appl. Phys., Vol. 25, No. 3, pp. 297-301, March, 1954.
11. W. S. Burnside and A. W. Panton, "The Theory of Equations", Vol. 1, Dover Publications, Inc., N. Y., 1960.
12. F. Cajori, "An Introduction to the Modern Theory of Equations", Macmillan Company, Inc., New York, 1904.
13. F. H. Effertz, "On the Synthesis of Networks Containing Two Kinds of Elements", Symposium Proceedings on Modern Network Synthesis, P.I.B., Vol. V, pp. 145-173, 1955.

SOME SYNTHESIS METHODS FOR ADJUSTABLE NETWORKS USING MULTI-VARIABLE TECHNIQUES¹

James F. Delansky

The Pennsylvania State University, University Park, Pa.

ABSTRACT

The synthesis methods developed here allow the designer 1) to choose a single type of adjustable building block for each independent real variable in the network description, 2) to maintain control of the functional complexity of these adjustable blocks, and 3) to use, in the single real variable case, only the absolute minimum number of such blocks.

INTRODUCTION

Adjustable networks are essentially networks with some elements whose values are adjustable and they may be either passive or active. The adjustable networks considered here are those which respond or adapt to an independent (with respect to frequency) real variable, or possibly several such independent real variables, and which for fixed values of these variables respond as linear time-invariant networks so that the usual steady-state frequency descriptions are valid. Adjustable networks with any number of ports can be considered. Some two-port examples are 1) an adjustable cutoff low-pass filter and 2) an adjustable group delay filter.

PRELIMINARIES

The heart of the synthesis procedure to be developed is an algebraic decomposition of matrix functions of two complex variables given by Youla². If $M(s, p)$ is a real rational matrix of the two complex variables s and p and is finite at $p = \infty$, then $M(s, p)$ has a representation as

$$M(s, p) = J(s) + H(s) (p I_\ell - F(s))^{-1} G(s) \quad (1)$$

where J , H , F , and G are rational matrices of appropriate orders. Furthermore, there is a smallest integer ℓ , say k , for which (1) holds. If an ℓ -port network η_ℓ terminates an $(n + \ell)$ -port coupling network η_c , the resulting n -port η is conveniently called a cascade-load connection. Let column matrices $\underline{a}$ and $\underline{b}$ represent port variables, excitation and response respectively. Then η_c and η_ℓ are described by

$$\underline{b}_c = N_c \underline{a}_c, \quad \underline{b}_\ell = N_\ell \underline{a}_\ell \quad (2)$$

For definiteness, the scattering description is chosen. Assume that η_ℓ consists of uncoupled one-ports and that the reflection coefficients ρ_i corresponding to each one-port are proportional. Then it can be shown that the scattering matrix of η is

$$N = N_{c11} + N_{c12} D (\rho^{-1} I_\ell - N_{c22} D)^{-1} N_{c21} \quad (3)$$

where D is a diagonal matrix of constants. Equations similar to (3) are obtained for immittance descriptions under the same assumptions on η_ℓ .

THE SYNTHESIS PROCEDURE USING ONE-PORT BUILDING BLOCKS

Suppose the prescribed data are a set of n positive numbers u_i and an $n \times n$ real rational scattering matrix $W(s, x)$ which describes an allowable n -port adjustable network,

$$W(s, x) = (B_q(x)s^q + \dots + B_o(x))/(b_m(x)s^m + \dots + b_o(x)) \quad (4)$$

where x is a real independent variable, the b 's are real polynomials, and the B 's are real $n \times n$ polynomial matrices. It is desired to synthesize (4), normalized to u_1 , using as adjustable elements only resistors of the form

$$R_j(x) = r_j (ax + b)/(cx + d) \quad (5)$$

where $r_j > 0, j = 1, 2, \dots, k$, and the constants a, b, c, d satisfy
 1) $ad - bc \neq 0$ and 2) either $a \neq 0$ or $c \neq 0$.

The reflection factor of the j^{th} adjustable resistor, normalized to $r_j > 0$, is

$$\rho_j = (a_1 x + b_1) / (c_1 x + d_1) \quad (6)$$

where the a_1, b_1, c_1, d_1 are linear combinations of a, b, c, d . Hence the collection of these uncoupled one-ports can be used for η_ℓ in the cascade load connection. Let $p = \rho^{-1}$ from which a unique solution for x is obtained as $x = (d_1 - b_1 p) / (a_1 p - c_1)$ and now let $\hat{W}(s, p) = W(s, (d_1 - b_1 p) / (a_1 p - c_1))$. Note a_1 and b_1 can always be chosen such that $\hat{W}(s, p)$ is finite at $p = \infty$. Then $\hat{W}(s, p)$ can be decomposed as in (1), which will identify η_c . Connecting η_c and η_ℓ in the cascade load connection then yields η as a realization of (4). Note that it is possible to obtain the minimum number of adjustable resistors of the form (5) needed in any synthesis of (4), and that control of the functional complexity of the adjustable blocks is attained in (5).

How is η to be synthesized? The following theorem is a key result¹: Necessary and sufficient conditions that (4) can be synthesized with the cascade load connection, where η_c is passive lossless and η_ℓ consists of decoupled adjustable resistors of the form (5) where k is the minimum possible integer, are that there exist values of a, b, c, d in (5) such that $\hat{W}(s, p)$ is 2-variable regular para-unitary. In general, η_c can be synthesized by active methods.

It can be shown that the synthesis procedure using adjustable resistors described in this section holds for general bilinear form adjustable 1-ports comprising η_ℓ , and modifications of the procedure allow bilinear form adjustable 2-ports to be used in η_ℓ . Extensions to the case of more than one independent real variable can also be shown¹.

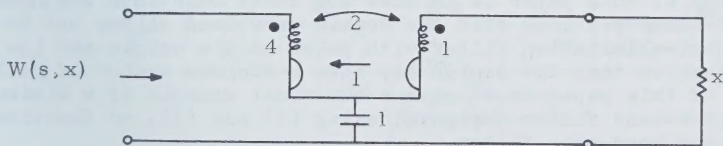
CONCLUSIONS

The major objection to most synthesis procedures for adjustable networks is that the adjustable elements used have had to be assumed highly versatile in their functional form. Also procedures that use more than one type of adjustable element are not desirable because of tracking problems. The procedure described here overcomes these difficulties.

EXAMPLE¹

Consider

$$W(s, x) = \frac{(4s^2 - s + 1)x + (-s^2 + 2s - 1)}{(4s^2 + s + 1)x + (s^2 + 2s + 1)} \quad (7)$$
 for which $a = d = 1, b = c = 0$ in (5) yields $\hat{W}(s, p)$ 2-variable regular para-unitary. The network is



REFERENCES

1. J. F. Delansky, "Some synthesis methods for adjustable networks using multi-variable techniques," Ph.D. dissertation, Cornell University, Ithaca, N.Y., June 1968.
2. D. C. Youla, P. Tissi, and W. Kohler, "The synthesis of networks containing lumped and distributed elements - parts I and II," Polytechnic Institute of Brooklyn, Technical Report No. RAD-66-489, October 1966.

M. SABLATASH

Dept. of Electrical Engineering,
The University of Toronto,
Toronto 5, Canada.

ABSTRACT

The determination of the constants d_1 and d_∞ in Watanabe's ideal transmission function, $w(\lambda)$, for a double pass-band filter is explained. The d_∞ is found by contour integration around the pass bands. The distribution of the total phase change between the two pass bands is discussed. A numerical example of a novel, double pass-band, minimum-inductance, insertion-loss filter, functioning as a band-stop filter, is presented. An iterative technique, based on Remez' algorithm, for obtaining $\phi(s)$ without the aid of $w(s)$ is also discussed briefly.

I. INTRODUCTION

Watanabe [1] has given the equation

$$w(\lambda) = \int_0^\lambda \left[d_\infty + \sum_{v=1}^m \frac{d_v}{\lambda^2 + Q_v^2} \right] \frac{d\lambda}{\sqrt{(\lambda^2 + k_1^2)(\lambda^2 + 1)}} \quad (1)$$

for the ideal transmission function of a double pass-band filter with pass bands in $[-jk_1, jk_1]$ and $[+j, \infty]$ in the λ -plane, and logarithmic singularities on the real and imaginary axes (so the Q_v^2 are negative or positive real numbers). The λ -plane is related to the usual s -plane, in which the pass bands are in $[\pm jk', \pm jk'_1]$ and $[+j, +j \frac{1}{k_2}]$, by

$$\lambda^2 = \frac{1 - k_2^2}{1 - k'^2} \frac{s^2 + k'^2}{k_2^2 s^2 + 1} \quad (2)$$

Watanabe's Corollary I [2] shows that the zeros of the characteristic function, $\phi(\lambda)$ (and, by (2), the zeros of $\phi(s)$), are the points at which

$$w(\lambda) = j(v + \frac{1}{2})\pi, \quad (3)$$

where v is an integer. Under certain restrictions [3] $w(\lambda)$ may be expressed as an algebraic function of λ , and it is not necessary to solve (3). For filters with two or three attenuation poles Watanabe [4] gives explicit, closed-form formulae for the restrictive condition and for $w(\lambda)$, and a numerical example using these formulae.

For cases in which the above-mentioned restrictions do not hold it appears that one must find the constants d_1 and d_∞ in (1) and solve (3). One of the aims of this paper is to show how these constants are found.

Watanabe [5] also mentions that the double pass-band filter may be useful as a band-elimination filter with poles at the origin and infinity, and implies that the design may have a minimum number of coils. A second aim of this paper is to give a numerical example of a minimum-coil double pass-band filter designed, using (1) and (3), to function as a narrow-band band-stop filter.

A third aim of this paper is to discuss the computational aspects of this design method, to suggest future possibilities in this regard, and to suggest an alternative computational method based on an algorithm developed by McGee [6].

II. DETERMINATION OF d_v AND d_∞

At the poles $\lambda = \pm jQ_v$, the residue of $dw/d\lambda$ is given by

$$\gamma_v = \frac{d_v}{2jQ_v \sqrt{(-Q_v^2 + k_1^2)(-Q_v^2 + 1)}}, \quad (4)$$

so

$$d_v = 2\gamma_v jQ_v \sqrt{(-Q_v^2 + k_1^2)(-Q_v^2 + 1)}. \quad (5)$$

The residues are restricted as follows.

From the principle of deformation of contours, the sums of the line integrals of $dw/d\lambda$ around the two pass bands (with no poles enclosed in the contour) equals 2π times the sum of the residues of $dw/d\lambda$ at all its poles. Therefore,

$$\oint_{\Gamma} \frac{dw}{d\lambda} d\lambda = 2\pi \sum_{v=1}^m 2\gamma_v, \quad (6)$$

where Γ denotes the two disjoint contours enclosing the two pass bands and no poles. The sum of the integrals of $w(j\omega)$ along the pass bands, in the positive direction and for positive frequencies, is one quarter of the value on the right side of (6); that is,

$$\Delta\beta_T = \pi \sum_{v=1}^m \gamma_v, \quad (7)$$

where $\Delta\beta_T$ is the change in phase of $w(\lambda)$. The sum of the two contour integrals given by (6) may be divided between the two pass bands so that the change in $\beta = \text{Im}[w(\lambda)]$ along each pass band in, say, the positive direction, and for positive frequencies, is an integral multiple of π . This is necessary to prevent a zero of

$$\varphi(\lambda) = \epsilon \cosh[w(\lambda)] \quad (8)$$

from appearing at a pass-band edge, which occurs when the change in β along a pass band is an odd multiple of $\pi/2$. Furthermore, a maximum or minimum of $\varphi(j\omega)$ must appear at any pass-band edge, so the total change along each pass band must be an integral multiple of $\pm\pi$. These restrictions lead to γ_v 's which are positive or negative integers, which are the values usually employed, or even numbers of γ_v 's which are odd multiples of $\pm\frac{1}{2}$, a possibility which has not been fully exploited yet.

The value of d_∞ is calculated from a contour integration. One first assigns a portion of $\Delta\beta_T$, given by (7), to each pass band, in such a way that each is an integral multiple of π . Thus, if $\Delta\beta_{T1}$ is the portion of $\Delta\beta_T$ assigned to the pass band contiguous to zero, and $\Delta\beta_{T2}$ is the portion assigned to the pass band contiguous to infinity,

$$\Delta\beta_T = \Delta\beta_{T1} + \Delta\beta_{T2}. \quad (9)$$

Then letting $v_v = Q_v$ for the h singularities on the imaginary λ -axis, and $u_\mu = Q_\mu$ for the singularities on the real λ -axis, one finds from (1) that

$$d_\infty = \frac{\Delta\beta_{T1} - \int_0^{k_1} \left[\sum_{v=1}^h \frac{d_v}{(v_v^2 - \omega^2)} + \sum_{\mu=h+1}^{m-h} \frac{d_\mu}{(u_\mu^2 + \omega^2)} \right] \frac{d\omega}{\sqrt{(k_1^2 - \omega^2)(1 - \omega^2)}}}{\int_0^{k_1} \frac{d\omega}{\sqrt{(k_1^2 - \omega^2)(1 - \omega^2)}}}, \quad (10)$$

or

$$d_\infty = \frac{\Delta\beta_{T2} + \int_1^\infty \left[\sum_{v=1}^h \frac{d_v}{(v_v^2 - \omega^2)} + \sum_{\mu=h+1}^{m-h} \frac{d_\mu}{(u_\mu^2 + \omega^2)} \right] \frac{d\omega}{\sqrt{(k_1^2 - \omega^2)(1 - \omega^2)}}}{\int_1^\infty \frac{d\omega}{\sqrt{(k_1^2 - \omega^2)(1 - \omega^2)}}}. \quad (11)$$

The various signs used have been determined by trial-and-error testing

a number of numerical examples.

The values of $\Delta\beta_{T1}$ and $\Delta\beta_{T2}$ used affect the attenuation ripple attained in each pass¹ band. Because they must each be integral multiples of π , only a relatively few choices are possible. With a little experience a choice yielding nearly equal ripples in both pass bands can usually be readily made. Often, $\Delta\beta_{T1} = \Delta\beta_{T2}$.

III. DERIVATION OF CHARACTERISTIC FUNCTIONS WITH A FACTOR $(s^2 - \sigma_0^2)$ FOR CRYSTAL-CAPACITOR DOUBLE PASS-BAND FILTERS WITH NEARLY EQUAL-RIPPLE PASS-BAND ATTENUATIONS

The residues, γ_v , of dw/ds at finite, non-zero imaginary-axis s -plane poles must be integers, so that no algebraic singularities result, and they may be positive for the most efficient approximation of attenuation specifications. The γ_∞ and γ_0 , corresponding, respectively, to s -plane poles at $s=\infty$ and $s=0$, must be $\pm\frac{1}{2}$, usually $\frac{1}{2}$ for [7] efficient approximation, and the γ_μ , the residue at $s=\pm\sigma_0$, must be -1 . Although other real-axis pole positions are rarely employed, it is to be noted that the sum of the residues at non-zero real-axis pole positions between $s=0$ and $s=\pm\sigma_0$ must be an integer, as must the sum of the residues at finite real-axis pole positions between $s=\pm\sigma_0$ and $s=\infty$. The residues at all these real-axis poles, excepting $s=\pm\sigma_0$, are usually positive. There may also be complex poles at which the residues, γ_i , may be integral multiples of $\pm\frac{1}{2}$; they are usually positive. The sum of the residues in all pole positions must be such that the total change in β in one direction along each pass band in the s -plane can be an integral multiple of π .

With the above-specified restrictions on the pole positions and the residues thereat, it follows from Watanabe's Lemma 1 that $w(s) = 0 + jk\frac{\pi}{2}$, where k is an odd integer, at two points, $+\sigma_1$ and $+\sigma_2$, on the real s -axis on either side of $-\sigma_0$, close to $+\sigma_0$ and $-\sigma_0$, respectively. Therefore, $\varphi(s)$, according to (8), is zero at these four points. Therefore, the perfectly equal-ripple $\varphi(s)$ has a factor

$$\frac{(s^2 - \sigma_1^2)(s^2 - \sigma_2^2)}{s^2 - \sigma_0^2} \quad . \text{ Since } \sigma_1 \approx \sigma_0 \approx \sigma_2, \text{ this factor may be replaced by } \frac{(s^2 - \sigma_0^2)^2}{s^2 - \sigma_0^2} = s^2 - \sigma_0^2, \text{ with little error in the attenuation characteristics.}$$

Since the zeros of the perfectly-ripple $\varphi(s)$ need not be found, σ_1 and σ_2 need not be calculated. All the other zeros of $\varphi(s)$ are found, in the usual case in which the only negative residues are at $s=\pm\sigma_0$, at the points along the pass bands where $w(s) = 0 + jk\frac{\pi}{2}$, where k is an odd integer. Usually the calculation is more efficiently done on the λ -plane. After these points have been found $\varphi(s)$ is found by multiplying its numerator polynomial, whose zeros are these points, by $(s^2 - \sigma_0^2)$. The pole positions are all those prescribed except those at $s=\pm\sigma_0$. The denominator polynomial of $\varphi(s)$ is one degree less than the numerator polynomial because $\gamma_\infty = \pm\frac{1}{2}$, and has the factor s because $\gamma_0 = \pm\frac{1}{2}$. Poles at finite, non-zero values of s with residues of $\pm\frac{1}{2}$, give rise to pole branch points of $\varphi(s)$, and those with residues which are odd integral multiples of ± 1 give rise to poles in $\varphi(s)$ of order equal to the magnitude of such residues.

A detailed numerical example of a crystal-capacitor double pass-band filter designed to function as a band-stop filter will illustrate the usefulness of such a $\varphi(s)$.

IV. DERIVATION OF CHARACTERISTIC FUNCTIONS WITH A FACTOR $(s+a)(s-b)$ FOR DOUBLE PASS-BAND FILTERS WITH NEARLY EQUAL-RIPPLE PASS-BAND ATTENUATION

Assuming that the residues at all poles--except those at $s=\pm\sigma_0$, at which positions no pole is assumed in this case--are the same as those in the preceding section, one places (as in the band-pass case) poles with residues of $-\frac{1}{2}$ (i.e., $\gamma_\mu = -\frac{1}{2}$) at two distinct pairs, $\pm\sigma_2$ and $\pm\sigma_3$, of σ -axis points. Then $\alpha=0$ at $s=\pm\sigma_1$ and $s=\pm\sigma_4$, where $0 < \sigma_1 \leq \sigma_2 \leq \sigma_3 \leq \sigma_4 < \infty$, and $\beta = k \frac{\pi}{2}$ (k an odd integer there). Therefore, there is a factor, $(s^2 - \sigma_1^2)(s^2 - \sigma_4^2)$, in the numerator of $\varphi(s)$. It would probably be rather troublesome (and it is not practically necessary) to calculate σ_1 and σ_4 , so one may assume that $\sigma_1 \approx \sigma_2$ and $\sigma_4 \approx \sigma_3$. The pole branch point of $\varphi(s)$ due to the factor $\sqrt{(s^2 - \sigma_2^2)(s^2 - \sigma_3^2)}$ in its denominator can then be cancelled, leaving the factor $\sqrt{(s^2 - \sigma_2^2)(s^2 - \sigma_3^2)}$ in the numerator. Then with $a=\sigma_2$, $b=\sigma_3$, and use of the fact that $\sqrt{(w^2+a^2)(w^2+b^2)} = |(s+a)(s-b)|_{s=jw}$, one obtains the desired $\varphi(s)$ with the factor $(s+a)(s-b)$. The factors $(s+a)(s+b)$ and $(s-a)(s-b)$ are not permissible, as is easily found by considering the possible resulting reactance functions. The parameters a and b are now available degrees of freedom for design. The attenuation characteristics should be negligibly disturbed by the approximations $\sigma_1 \approx \sigma_2$ and $\sigma_4 \approx \sigma_3$.

V. BRIEF CONSIDERATION OF REALIZABILITY AND REALIZATION PROCEDURES

If there is a factor $(s^2 - \sigma_0^2)$ in the numerator of $\varphi(s)$, the open- and short-circuit reactances are, respectively,

$$X_{10} = X_{20} = \frac{E_o}{E_e \pm F_e} \quad (12)$$

and

$$X_{1s} = X_{2s} = \frac{E_e \mp F_e}{E_o}, \quad (13)$$

where E_e =even part of $E(s)$ and E_o =odd part of $E(s)$. It should be noted that cancellations of powers of s occur, and that the network must be realized from both ends to find all the capacitors. After realization, simple network transformations with T 's, π 's and L 's of capacitors result in the desired crystal (or ceramic) and capacitor ladder network.^[9] The realization can obviously be effected by a crystal-capacitor lattice network.

VI. NUMERICAL DESIGN EXAMPLE OF CRYSTAL-CAPACITOR DOUBLE PASS-BAND FILTER

A novel double pass-band filter with no coils is designed to function as a band-stop filter by making the lower and upper pass bands quite wide, the centre stop band very narrow, and by using attenuation poles of the lowest possible order at $s=0$ and $s=\infty$ in the lowest and highest stop bands, respectively. The lower pass band is specified to be between the limits of 60 khz and 104.025 khz and the upper pass band between 104.135 khz and 108.000 khz. Poles with associated γ 's equal to unity at finite, non-zero frequencies, and equal to $\frac{1}{2}$ at $f=0$ and $f=\infty$ are specified at $f_0=0$, $f_1=104.075$ khz, $f_2=104.085$ khz and at infinity. Frequency normalization with respect to the lower edge of the upper pass band yields a lower pass band in $[jk', jk_1] = [j0.57617515, j0.99894367]$, and in $[-jk_1', -jk']$, an upper pass band in $[j1, j(1/k_2)] = [j1.00, j1.03711528]$ and in $[-j(1/k_2), -j1]$, and

imaginary-axis poles at $s_0=0$, $s_1=\pm j0.9994238$, $s_2=\pm j0.999519854$, and $s_3=\infty$. By use of

$$\lambda^2 = \frac{1-k^2}{1-k'^2} \frac{s^2+k'^2}{k^2 s^2+1}, \quad (14)$$

these specifications are transformed to the λ -plane. Equation (15) becomes

$$\lambda^2 = 0.1052260815 \frac{s^2+0.3319778034775225}{0.9297066418740912s^2+1}. \quad (15)$$

A real pole with residue -1 is also located in the λ -plane corresponding to the imaginary frequencies $f_R=\pm j65.39538$ khz. Since $2Y_0=2Y_\infty=Y_1=Y_2=1$, the sum of contour integrals around both pass bands is $6 \times 2\pi-4\pi=8\pi$. The integral along each pass band (in one direction) for, say, positive frequency only is, therefore, equal to π . Equation (5) is used to calculate the values of d_v and d_u in (1). Using subscripts corresponding to those of the prescribed poles, these are found to be: $d_0=0.19058473$, $d_1=0.02982497$, $d_2=0.02994296$, $d_3=0.36938026$ and $d_4=-0.56985524$. The poles in the λ -plane defined by (15) are at $\pm\lambda_0=u_0=\pm 0.18690299$, $\pm\lambda_1=\pm jv_1=\pm j0.99161139$, $\pm\lambda_2=\pm jv_2=\pm j0.99299642$, $\pm\lambda_3=\pm u_3=\pm 0.33642539$ and $\pm\lambda_4=\pm u_4=\pm 0.26946530$. The ideal transmission function in the λ -plane is known except for d_∞ and is given by:

$$w(\lambda) = \int_0^\lambda \left[d_\infty + \frac{-0.19058473}{-\lambda^2+(0.18690299)^2} + \frac{-0.02982497}{\lambda^2+(0.99161139)^2} + \frac{-0.02994296}{\lambda^2+(0.99299642)^2} + \frac{-0.36938026}{-\lambda^2+(0.33642539)^2} + \frac{0.56985524}{-\lambda^2+(0.26946530)^2} \right] \frac{d\lambda}{-\sqrt{[\lambda^2+(0.98476198)^2]} [\lambda^2+1]}. \quad (16)$$

Since the total change in β along the mapping of each pass band to the λ -plane, in one direction, is π ,

$$\pi = \int_0^{0.98476198} \left[d_\infty + \frac{-0.19058473}{w^2+(0.18690299)^2} - \frac{0.02982497}{(0.99161139)^2-w^2} - \frac{0.02994296}{(0.99299642)^2-w^2} + \frac{-0.36938026}{w^2+(0.33642539)^2} + \frac{0.56985524}{w^2+(0.26946530)^2} \right] \frac{dw}{(-1)\sqrt{[1-w^2]} [(0.98476198)^2-w^2]}. \quad (17)$$

Solution for d_∞ from (17) yields

$$d_\infty = \frac{0.98476198 \int_0^{0.98476198} \left[\frac{0.19058473}{w^2+(0.18690299)^2} + \frac{0.02982497}{(0.99161139)^2-w^2} + \frac{0.02994296}{(0.99299642)^2-w^2} + \frac{0.36938026}{w^2+(0.33642539)^2} + \frac{-0.56985524}{w^2+(0.26946530)^2} \right] \frac{dw}{\sqrt{[1-w^2]} [(0.98476198)^2-w^2]}}{0.98476198 \int_0^{0.98476198} \frac{dw}{-\sqrt{[1-w^2]} [(0.98476198)^2-w^2]}}. \quad (18)$$

The signs used for the various factors in (17) were deduced by trial and error and by checking the values of d 's obtained for a two-pole case using Watanabe's Theorem 6.^[10] Evaluation of the right side of (18) using Simpson's rule with decreased step sizes near the branch points yields $d_\infty = -0.02313212$. This is checked by integration along

the upper pass band, for which the square root sign in the denominator is to be taken positive rather than negative. The expression for d_∞ for this case is

$$\pi - \int_1^\infty \left[\frac{-0.19058473}{w^2 + (0.18690299)^2} + \frac{0.02982497}{w^2 - (0.99161139)^2} + \frac{0.02994296}{w^2 - (0.99299642)^2} - \frac{0.36938026}{w^2 + (0.33642539)^2} + \frac{0.56985524}{w^2 + (0.26946530)^2} \right] \frac{dw}{\sqrt{[1-w^2] [(0.98476198)^2 - w^2]}}, \quad (19)$$

and the evaluation of the right side again yields $d_\infty = -0.02313212$. The points along the pass bands where β is an odd multiple of $\pi/2$ are calculated next by evaluating $\beta(w)$ and then an iterative procedure is used to obtain these points to a high degree of accuracy. The equation for $\beta(w)$ used in the mapping of the lower pass band to the λ -plane is

$$\beta(w) = \int_0^w \left[0.02313212 + \frac{0.19058473}{w^2 + (0.18690299)^2} + \frac{0.02982497}{(0.99161139)^2 - w^2} + \frac{0.02994296}{(0.99299642)^2 - w^2} + \frac{0.36938026}{w^2 + (0.33642539)^2} - \frac{0.56985524}{w^2 + (0.26946530)^2} \right] \frac{dw}{\sqrt{[1-w^2] [(0.98476198)^2 - w^2]}}, \quad (20)$$

and for the mapping of the upper pass band to the λ -plane is

$$\beta(w) = \int_1^w \left[-0.02313212 + \frac{-0.19058473}{w^2 + (0.18690299)^2} + \frac{0.02982497}{w^2 - (0.99161139)^2} + \frac{0.02994296}{w^2 - (0.99299642)^2} + \frac{-0.36938026}{w^2 + (0.33642539)^2} + \frac{0.56985524}{w^2 + (0.26946530)^2} \right] \frac{dw}{\sqrt{[1-w^2] [(0.98476198)^2 - w^2]}}. \quad (21)$$

Essentially, the square roots are of different sign in (20) and (21).

It is found, from (20), that $\beta = \pi/2$ on the λ -plane imaginary axis at $w = 0.98166560$ and from (21) that $\beta = \pi/2$ on the λ -plane imaginary axis at $w = 1.00306781$. These zeros of the characteristic function are mapped to the s -plane imaginary axis using (15) to obtain the positions of the imaginary-axis zeros of $\varphi(s)$. The imaginary-axis zeros of $\varphi(s)$ in the s -plane are at $s = \pm j0.99872369$ and at $s = \pm j1.00020751$. Using the mappings of the λ -plane real-axis pole pair positions as the real-axis zeros of $\varphi(s)$ the characteristic function without a constant multiplier to adjust the maximum attenuation in the pass bands is

$$\varphi_1(s) = \frac{(s^2 - 0.99894395695)(s^2 + 0.9974489995)(s^2 + 1.000415084)}{s(s^2 + 0.998847979978783)(s^2 + 0.99903993854018)}. \quad (22)$$

A constant multiplier, K , is found such that the operating (transducer) loss at the pass-band edge $s = j0.99894367$ is 0.25 dB, using the equation

$$\alpha = 10 \log_{10} \left[1 + \frac{1}{K^2} \varphi_1^2(jw) \right]. \quad (23)$$

This results in the characteristic function

$$\varphi(s) = \frac{F(s)}{P(s)} = \frac{s^6 + 0.9989201265431088600847s^4 - 0.9978912283978705544636s^2 - 0.99680923830302411665764}{9.489045036052677496152s^5 + 18.95804843581198206016s^3 + 9.469013894738226092125s} \quad (24)$$

Because $\varphi(s)$ is an odd function of s , $E(s)$, the numerator of the operating voltage ratio, is found by finding the roots of $F(s)+P(s)$, negating those with positive real parts, and then multiplying all the resulting factors--corresponding to the roots of $F(s)+P(s)$ with negative real parts, and those with positive real parts that are negated--to obtain $E(s)$. A ladder realization without negative elements could not be obtained, so a lattice is realized.^[11] To this end the factors of $E(s)$ corresponding to roots of $F(s)+P(s)$ with negative real parts are multiplied together to obtain $E_{\alpha}(s)$, a third degree polynomial. The remaining factors of $E(s)$, corresponding to the roots of $F(s)+P(s)$ with positive real parts, are multiplied together to obtain $E_{\beta}(s)$, another third degree polynomial. With, for example,

$$E_{\alpha}(s) = E_{\alpha e}(s) + E_{\alpha o}(s), \quad (25)$$

where $E_{\alpha e}(s)$ and $E_{\alpha o}(s)$ are the even and odd parts of $E_{\alpha}(s)$, respectively, the horizontal arm of the impedance- and frequency-normalized lattice is realized from^[12]

$$z_b = \frac{E_{\alpha e}(s)}{E_{\alpha o}(s)}, \quad (26)$$

and the cross arms from

$$z_a = \frac{E_{\beta e}(s)}{E_{\beta o}(s)}. \quad (27)$$

After denormalization to $R_0=135$ ohms and $f_0=104.135$ khz, the lattice network shown in Figure 1(a) and the equivalent economy bridge network shown in Figure 1(b) are obtained.^[13] The operating (transducer) loss versus frequency is given by Figure 2, which shows that the pass-band ripple is less than 0.25 dB, and almost equi-ripple, and that the attenuation rises slowly as $f \rightarrow 0$ and $f \rightarrow \infty$.

By changing the steepness of the stop-band skirts, the pole separations, and pass-band limits, considerable variations in element values can be obtained. To the writer's best knowledge, this design of a band-stop filter is original: it is not to be found in the earlier literature.

VII. AN ALTERNATE METHOD FOR FINDING $\varphi(s)$

The calculations of $\varphi(s)$ for the example of the previous section took about 10 minutes on an IBM 7094. The calculation of the integrals with pole branch points is fairly time-consuming. This makes one wonder why a much faster method for evaluating integrals has not yet been found. Perhaps some device--using, for example, light to produce a fine-resolution curve trace--which finds the area under a curve should be invented.

McGee has used a Remez-type algorithm (whose convergence was not proved by McGee), to obtain a $\varphi(s)$ with all its zeros and poles on the $s=j\omega$ axis.^[14] By using the weighting factor $(s^2-\sigma_0^2)$ in an initial $\varphi(s)$ (i.e., $\omega^2+\sigma_0^2$ in $\varphi(j\omega)$), one should be able to implement the algorithm for the determination of a $\varphi(s)$ for double pass-band filters with this factor. One could clearly obtain a $\varphi(s)$ with the factor $(s-a)(s+b)$ in a similar way. This has not been tried on an example, but no diffi-

culty is apparent.

It is to be noted that Schumacher has recently found a convergence proof for the algorithm proposed by McGee, although he does not deal with the problem of including real zeros.^[15]

ACKNOWLEDGMENTS

The author is grateful to the Northern Electric Co. R.&D. Labs., Ottawa, for providing computational facilities, and Mrs. Endrenyi of IBM, Toronto, who wrote the computer programs for finding d_z , d_v , and the zeros of $\varphi(s)$. The author is most grateful to Prof. T.J. Higgins, his Ph.D. thesis advisor, for encouragement in finally solving some of the mathematical niceties of the techniques, for which the author had used the potential analogy previously.

REFERENCES

- [1] H. Watanabe, "Approximation Theory for Filter Networks," IRE Trans. on Circuit Theory, CT-9, pp. 341-56, September, 1961.
- [2] Ibid., p.343.
- [3] Ibid., pp. 348-51.
- [4] Ibid., p. 351.
- [5] Ibid., p. 350.
- [6] W.F. McGee, "Numerical Approximation Technique for Filter Characteristic Functions," IEEE Trans. on Circuit Theory, CT-14 (Correspondence), pp. 92-4, March, 1967.
- [7] M. Sablatash, "Determination of Characteristic Functions for Crystal-Capacitor Band-Pass Ladder Filters with Nearly Chebyshev Pass-Band Attenuation," Proc. of the Second Annual Allerton Conf. on Circuit and System Theory, The University of Illinois, Urbana, Ill., pp. 149-74, 1964.
- [8] R. Saal and E. Ulbrich, "On the Design of Filters by Synthesis," IRE Trans. on Circuit Theory, CT-5, pp. 297 and 309-13, December, 1958.
- [9] E.A. Guillemin, Synthesis of Passive Networks, New York: Wiley, 1957, pp. 150-7.
- [10] Watanabe, op. cit., pp. 348-51.
- [11] W. Cauer, Synthesis of Linear Communication Networks. New York: McGraw-Hill, 1958, pp. 434-43.
- [12] Ibid., p. 443.
- [13] Ibid., p. 112.
- [14] McGee, loc. cit.
- [15] L. Schumacher, Unpublished work, 1968, Mathematics Research Centre, The University of Wisconsin, Madison, Wis. The proof has been submitted for publication in the IEEE Trans. on Circuit Theory.

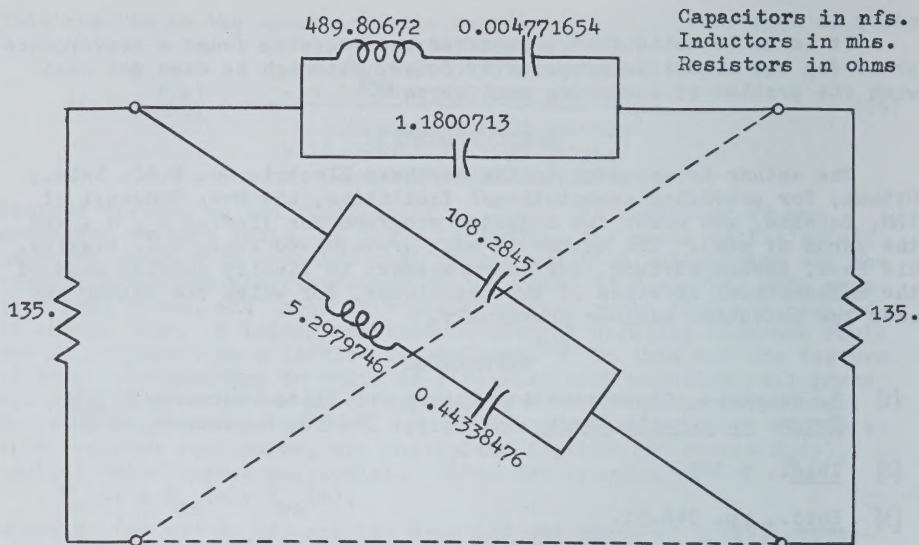


Figure 1(a) Double Pass-Band Filter with only Crystals and Capacitors Serving as a Narrow-Band Band-Stop Filter

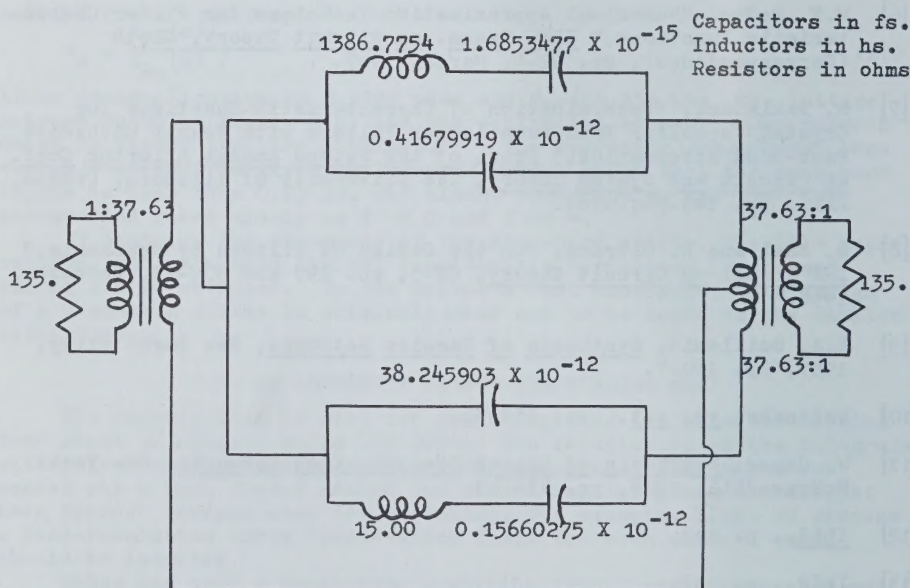


Figure 1(b) An Economy Bridge Equivalent of the Lattice Filter in Figure 1(a)

DETAILS OF PASS BANDS

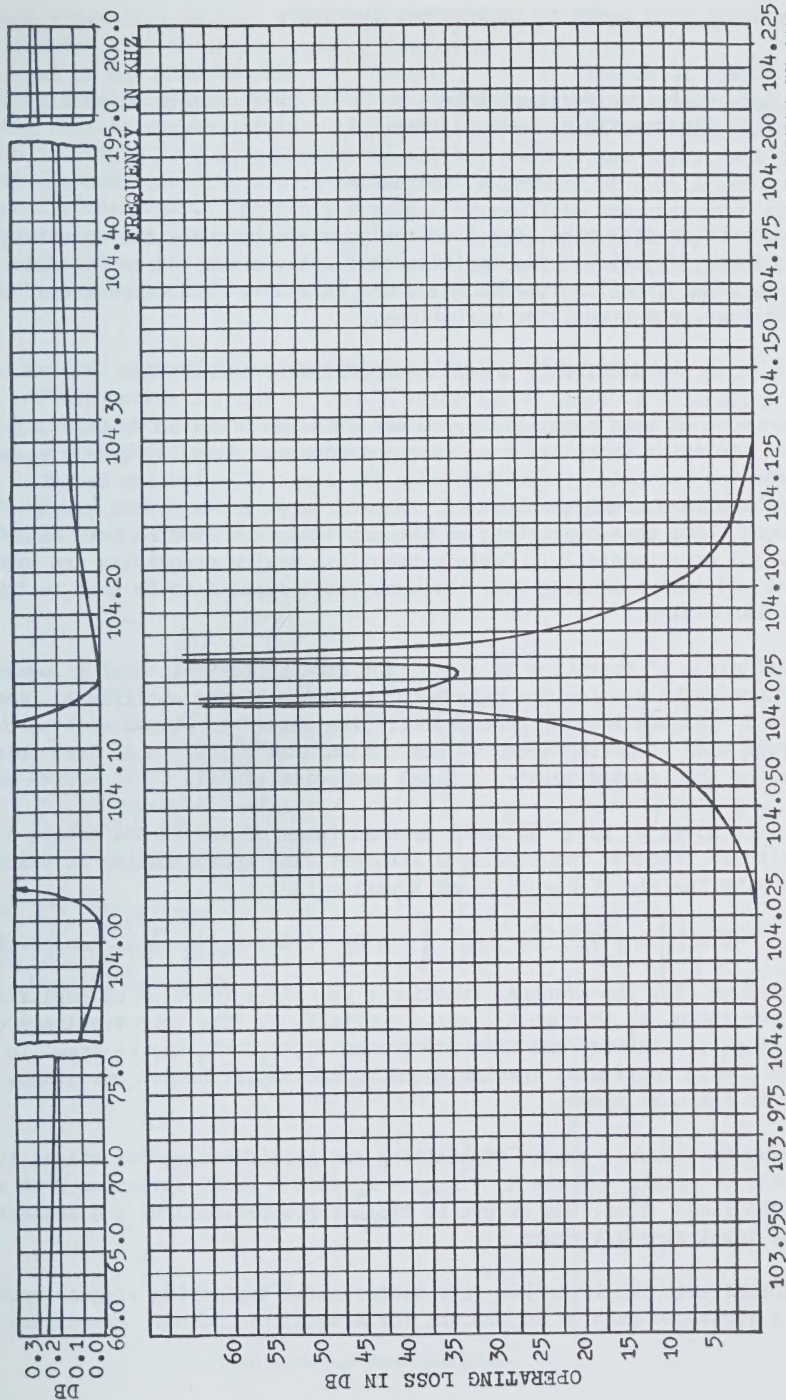


Figure 2 Operating Attenuation of the Double Pass-Band Filters in Figures 1(a) and 1(b)

A LOW LEVEL WIDE RANGE AC LOGARITHMIC AMPLIFIER

ROBERT S. CUIKAY

A.F. Cambridge Research Labs.

L.G. Hanscom Field, Bedford, Mass.

R.E. SPRAQUE and J. FAY

Northeastern University

Boston, Mass.

ABSTRACT

An analysis is presented of a low level, constant band-width ac voltage logarithmic amplifier. The amplifier has a logarithmic transfer function over an input range of five decades, and is stable with temperature, supply voltage, and transistor variations.

DESCRIPTION OF THE AC LOGARITHMIC AMPLIFIER

The purpose of this paper is to present a design of an ac voltage logarithmic amplifier whose transfer function covers a five (5) decade range and remains essentially constant over the range of operation despite variations in ambient temperature, transistor parameters, and supply voltages. The characteristics of the amplifier are studied both analytically and experimentally. Certain practical design principles are presented and analyzed along with the analytical results to be used in the amplifier design.

The principle of operation is a straight line approximation of a logarithmic function by summing the outputs of clipping linear amplifiers, Curve A, Figure 1. Figure 2 shows cascaded clipping amplifier stages with an output from each stage connected to one common load point. Each load point signal is then summed to give a final composite signal.

The basic model to cover an input dynamic range of 100 db and create a logarithmic transfer function by a straight line approximation is calculated with the aid of Figure 2 and Equation (1).

$$e_o/e_{in} = k A_1 \left\{ 1 + A_2 (1 + A_3 [1 + A_4 ((1 + A_5))]) \right\} \quad (1)$$

In equation (1), the transfer function e_o/e_{in} is a function of amplifier gain per stage, A_1 through A_5 , and a constant, k . The output voltage of each stage is limited, and thus the curves $A_1, A_2, \dots, A_5$ may be drawn in on Figure 1 with horizontal spacing equal to the amplification per stage, 20 db in the case shown.

Choosing the correct amount of limiting and amplification per stage, it is possible to obtain a curve $A_1 + A_2 + \dots + A_5$ with a small deviation from a straight line. Thus, the output is closely proportional to the logarithm of the input signal (Figure 3).

The logarithmic amplifier has five differential amplifying stages, each with a (voltage) gain of 10 (20 db) for a total of 100 db. Associated

with each stage is a voltage limiting network and a summing resistance which operates into an active summing network.

The signal from each amplifying point of similar phase is summed through a summing resistance and fed into the emitter of a common-base amplifier. As shown in Figure 2, the collectors of all the common-base amplifiers in the active summer circuits are connected to the primary of a balanced-to-unbalanced transformer. The final result (Figure 3) is the summation of all signals leaving the signal points of the compression amplifier. The output signal from the secondary of the summing transformer is applied to the synchronous detector for signal detection and ac to dc voltage conversion.

The synchronous detector converts the logarithmic signal voltage into an analog voltage having a range of 0 to 5 volts for an input of 5 decades. Figure 3 is a plot of the synchronous detector output voltage as a function of the logarithmic amplifier input.

Over an ambient temperature range from -10°C to $+50^{\circ}\text{C}$, the temperature stability achieved is a value less than $\pm 5\%$ of the full scale output voltage.

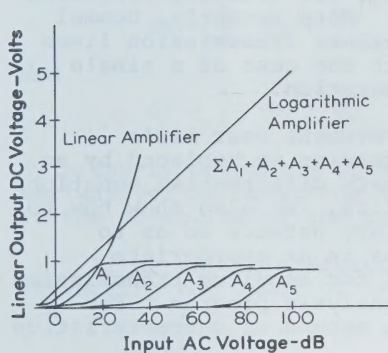


Figure 1

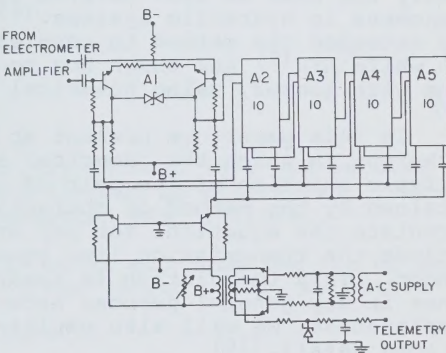
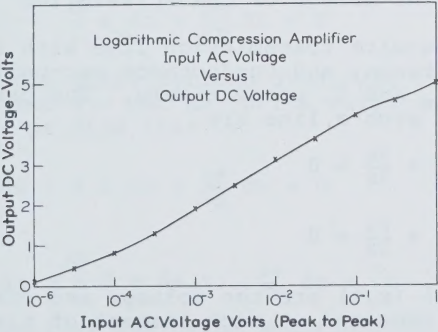


Figure 2



TRANSIENT ANALYSIS OF LOSSY TRANSMISSION LINES

FRANKLIN H. BRANIN, JR.
IBM Systems Development Laboratory
Kingston, New York

ABSTRACT

The method of characteristics is extended to lossy transmission lines using an improvement of Liu's technique. Instead of numerical integration of the ordinary differential equations along the forward and backward characteristics, an analytic solution is developed. In addition, a general method of formulation is described which enables the transmission line equations to be coupled to those of an arbitrary network.

INTRODUCTION

Transient analysis of transmission lines, which is of increasing importance in computer-aided design of electronic circuits, can be effected rapidly using the method of characteristics. This method converts the two partial differential equations of the transmission line into a pair of ordinary differential equations which are easily solved. The solution for lossless lines described by Branin⁽¹⁾ is similar to that proposed earlier by Pendred⁽²⁾ and others⁽³⁾⁻⁽⁵⁾ and is essentially the same as that derived long ago for treating transient phenomena in hydraulic systems.^{(6),(7)} More recently, Dommel has extended the method to coupled lossless transmission lines⁽⁸⁾ while Liu⁽⁹⁾ has shown how to treat the case of a single line with losses, using numerical integration.

In this paper, we present an improvement over Liu's technique in which the numerical integration is replaced by an analytic solution of the pair of ordinary differential equations obtained by the method of characteristics. We also show how to formulate the equations for any arbitrary network so as to include the transmission line equations in an appropriate manner. This formulation is essential for handling transmission lines in any general purpose network analysis program. For completeness, we will also explain the method of characteristics in some detail.⁽¹⁰⁾

THE METHOD OF CHARACTERISTICS^{(11),(12)}

Consider a two-wire transmission line with its resistance, inductance, capacitance, and conductance per-unit-length designated as R , L , C , and G , respectively. The partial differential equations for such a line are

$$L \frac{\partial i}{\partial t} + Ri + \frac{\partial e}{\partial x} = 0 \quad (1)$$

$$C \frac{\partial e}{\partial t} + Ge + \frac{\partial i}{\partial x} = 0 \quad (2)$$

where $e(x,t)$ and $i(x,t)$ are the voltage and current at each point x along the line and at each instant of time t .

The method of characteristics for solving these equations is based on a transformation in the x - t plane which accomplishes the conversion of Equations (1) and (2) into a pair of ordinary differential equations. Each of these two ordinary differential equations holds true along a different family of characteristic curves in the x - t plane, one family corresponding to the forward or incident wave and the other to the backward or reflected wave. These two families of characteristic curves are, in effect, the basis for a new coordinate system instead of the canonical coordinates $x=\text{constant}$ and $t=\text{constant}$.

The transformation involved in the method of characteristics uses the equations

$$\frac{dx}{dt} = + 1/\sqrt{LC} \quad (3)$$

and

$$\frac{dx}{dt} = - 1/\sqrt{LC} \quad (4)$$

which define the characteristic curves. When L and C are constant, as we will assume they are, the characteristic curves are straight lines and the phase velocity, $1/\sqrt{LC}$, is constant. Since both e and i are functions of x and t , their total differentials are

$$de = \frac{\partial e}{\partial x} dx + \frac{\partial e}{\partial t} dt \quad (5)$$

and

$$di = \frac{\partial i}{\partial x} dx + \frac{\partial i}{\partial t} dt \quad (6)$$

By properly combining Equations (1) through (6), the partial derivatives in Equations (1) and (2) can be eliminated. This is the objective of the transformation that follows.

If we multiply Equation (1) by dx and Equation (2) by dt/C (reversing the order of terms in the latter case), we obtain

$$L \frac{\partial i}{\partial t} dx + R i dx + \frac{\partial e}{\partial x} dx = 0 \quad (7)$$

and

$$\frac{1}{C} \frac{\partial i}{\partial x} dt + \frac{G}{C} e dt + \frac{\partial e}{\partial t} dt = 0 \quad (8)$$

If we then make the substitutions $dx=dt/\sqrt{LC}$ in the first term of Equation (7) and $dt=\sqrt{LC} dx$ in the first and second terms of Equation (8), we find that

$$\sqrt{\frac{L}{C}} \frac{\partial i}{\partial t} dt + R i dx + \frac{\partial e}{\partial x} dx = 0 \quad (9)$$

$$\sqrt{\frac{L}{C}} \frac{\partial i}{\partial x} dx + \sqrt{\frac{L}{C}} G e dx + \frac{\partial e}{\partial t} dt = 0 \quad (10)$$

Adding these equations and using Equations (5) and (6) yields the result

$$\sqrt{\frac{L}{C}} \frac{di}{dt} + \left(R i + \sqrt{\frac{L}{C}} G e \right) dx + de = 0 \quad (11)$$

which holds true only along the forward characteristics, defined as $dx/dt = 1/\sqrt{LC}$.

If we make similar substitutions in Equations (7) and (8) using the relations $dx = -1/\sqrt{LC}$ and $dt = -\sqrt{LC} dx$, we find that

$$-\sqrt{\frac{L}{C}} \frac{di}{dt} + \left(R i - \sqrt{\frac{L}{C}} G e \right) dx + de = 0 \quad (12)$$

which holds true only along the backward characteristics, defined as $dx/dt = -1/\sqrt{LC}$. For lossless transmission lines, where $R=0$ and $G=0$, these equations become

$$de = -Z_0 di \quad (13)$$

and

$$de = +Z_0 di \quad (14)$$

where $Z_0 = \sqrt{L/C}$ is the characteristic impedance of the line. Equations (13) and (14) can be integrated directly, yielding exact solutions of the form

$$\Delta e = -Z_0 \Delta i \quad (15)$$

and

$$\Delta e = +Z_0 \Delta i \quad (16)$$

These equations, which were also used by the authors previously cited, (2)-(8) hold true along the forward and backward characteristics, respectively.

Solution for Transmission Line with Losses

Liu cast Equations (11) and (12) into a particularly simple and useful form by differentiating them with respect to time and then invoking Equations (3) and (4) to eliminate dx/dt ; the results so obtained, after some rearrangement, are

$$\frac{de}{dt} + Z_0 \frac{di}{dt} = - \left(\frac{G}{C} e + \frac{R}{L} Z_0 i \right) \quad (17)$$

which holds true along the forward characteristics, $dx/dt = 1/\sqrt{LC}$, and

$$\frac{de}{dt} - Z_0 \frac{di}{dt} = - \left(\frac{G}{C} e - \frac{R}{L} Z_0 i \right) \quad (18)$$

which holds true along the backward characteristics, $dx/dt = -1/\sqrt{LC}$. We now make the substitutions

$$y = e + Z_0 i \quad (19)$$

and

$$z = e - Z_0 i \quad (20)$$

from which it follows that

$$\frac{G}{C} e \pm \frac{R}{L} z_0 i = \frac{G}{C} \frac{(y+z)}{2} \pm \frac{R}{L} \frac{(y-z)}{2} \quad (21)$$

Accordingly, Equations (17) and (18) may be written in matrix form as

$$\begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} (G/C + R/L) & (G/C - R/L) \\ (G/C - R/L) & (G/C + R/L) \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \quad (22)$$

Now Liu, after asserting that Equations (17) and (18) or, equivalently Equation (22) -- are not directly integrable, points out that they can be solved by numerical integration, e.g. the Runge-Kutta method. The present author also claimed that Equations (11) and (12) were not integrable. But it is now evident that direct integration can indeed be carried out as follows:

If Equation (22) is written in the simpler form $\dot{u}=Au$, an analytic solution can be found in terms of the eigenvalues and eigenvectors of the coefficient matrix, A .⁽¹³⁾ Let X be a square matrix whose columns are the eigenvectors of A and Λ a diagonal matrix of the eigenvalues of A . Then A may be diagonalized by the transformation

$$X^{-1} A X = \Lambda \quad (23)$$

This transformation also diagonalizes the transition matrix e^{At} thus:

$$X^{-1} e^{At} X = e^{\Lambda t} \quad (24)$$

which may be solved for e^{At} yielding

$$e^{At} = X e^{\Lambda t} X^{-1} \quad (25)$$

The solution for $\dot{u}=Au$, then, may be expressed as

$$u(t+h) = e^{Ah} u(t) = X e^{\Lambda h} X^{-1} u(t) \quad (26)$$

For the coefficient matrix of Equation (22), the desired diagonalizing transformation turns out to be (when $G/C \neq R/L$)

$$-\frac{1}{2} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \left(\frac{G}{C} + \frac{R}{L}\right) & \left(\frac{G}{C} - \frac{R}{L}\right) \\ \left(\frac{G}{C} - \frac{R}{L}\right) & \left(\frac{G}{C} + \frac{R}{L}\right) \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = - \begin{bmatrix} \frac{G}{C} & 0 \\ 0 & \frac{R}{L} \end{bmatrix} \quad (27)$$

Then using Equations (25) and (26) along with

$$e^{\Lambda} = \begin{bmatrix} e^{-G/C} & 0 \\ 0 & e^{-R/L} \end{bmatrix} \quad (28)$$

we find that

$$\begin{bmatrix} y(t+h) \\ z(t+h) \end{bmatrix} = \begin{bmatrix} e^{-hG/C} + e^{-hR/L} & (e^{-hG/C} - e^{-hR/L}) \\ (e^{-hG/C} - e^{-hR/L}) & (e^{-hG/C} + e^{-hR/L}) \end{bmatrix} \begin{bmatrix} y(t) \\ z(t) \end{bmatrix} \quad (29)$$

To be precise, y and z are functions of both x and t . But since $y(x, t)$ is "confined" to the forward characteristic, any variation in y is governed not only by Equations (22) and (29) but also by the constraint $dx=dt/\sqrt{LC}$; similarly, variations in $z(x, t)$ are subjected to the constraint $dx=-dt/\sqrt{LC}$. Accordingly, if we wish to solve Equation (22) at some point $(x, t+h)$, we must be more careful and write Equation (29) in the form

$$\begin{bmatrix} y(x, t+h) \\ z(x, t+h) \end{bmatrix} = \begin{bmatrix} (e^{-hG/C} + e^{-hR/L}) & (e^{-hG/C} - e^{-hR/L}) \\ (e^{-hG/C} - e^{-hR/L}) & (e^{-hG/C} + e^{-hR/L}) \end{bmatrix} \begin{bmatrix} y(x-h/\sqrt{LC}, t) \\ z(x+h/\sqrt{LC}, t) \end{bmatrix} \quad (30)$$

since when time changes from t to $t+h$, the space variable x changes from $x-h/\sqrt{LC}$ to x along the forward characteristic and from $x+h/\sqrt{LC}$ to x along the backward characteristic. (See Figure 1.)

Using Equations (19), (20), and (30), then, we finally obtain

$$e(x, t+h) = \frac{1}{2} e^{-hG/C} \left[e(x-dx, t) + e(x+dx, t) + Z_0 i(x-dx, t) - Z_0 i(x+dx, t) \right] \quad (31)$$

and

$$Z_0 i(x, t+h) = \frac{1}{2} e^{-hR/L} \left[e(x-dx, t) - e(x+dx, t) + Z_0 i(x-dx, t) + Z_0 i(x+dx, t) \right] \quad (32)$$

where $dx=h/\sqrt{LC}$. For the distortionless line, where $G/C=R/L$, the variables y and z in Equation (22) are no longer interdependent and so the mechanism of Equations (23) to (28) is unnecessary. Direct integration may be used instead and the resulting equations solved for $e(x, t+h)$ and $Z_0 i(x, t+h)$, yielding Equations (31) and (32) again.

Thus, Equations (31) and (32) represent an exact solution for the voltage and current at a finite set of equidistant points along the line and at a finite set of points in time. The incident voltage or current wave, moreover, may be identified as the set of values of e or i along a forward characteristic. Similarly, the reflected wave is given by the set of values of voltage or current along a backward characteristic.

Formulation of Equations with Terminating Networks

We now need a general method of formulating the terminal equations of the transmission line in conjunction with the equations for whatever terminating networks are involved. For

this purpose, we will modify the mixed or hybrid method of formulation described by Branin. (13)

At the input of the transmission line Equation (18), which holds along the backward characteristic, may be rearranged (and multiplied by $C Z_0$) to give

$$Z_0 \left(C \frac{de}{dt}(0,t) + G e(0,t) \right) = L \frac{di}{dt}(0,t) + R i(0,t) \quad (33)$$

This equation we interpret as describing the relation between voltage and current in a special device or equivalent network shown at the left in Figure 2b. At the output, we obtain from Equation (17) a similar relation

$$Z_0 \left(C \frac{de}{dt}(d,t) + G e(d,t) \right) = -L \frac{di}{dt}(d,t) - R i(d,t) \quad (34)$$

Notice that the direction of current flow in the transmission line is opposite to that of the conventional current flow in the output equivalent network of Figure 2b. For convenience, however, in treating both the input and output equivalent networks uniformly, we will not explicitly display the negative sign for the current in the latter. Hence, we may abbreviate Equations (33) and (34) to read

$$Z_0 Y e = Z i \quad (35)$$

where $Y = Cd/dt + G$ and $Z = Ld/dt + R$, or alternatively,

$$Y_0 Z i = Y e \quad (36)$$

where $Y_0 = \sqrt{C/L} = 1/Z_0$.

Extending the mixed or hybrid method of analysis (13), we consider not only y-branches (admittances) and z-branches (impedances) but also x-branches, which are the transmission line equivalent networks shown in Figure 2b. These x-branches will be connected in place of the corresponding input or output terminals of each transmission line.

If we order the branches in the sequence y, x, z and carry out a tree-link sort, we can then write the generalized (hybrid) form of Ohm's law for the entire network (see Equation (30) of reference 13) as follows:

$$\begin{bmatrix} U_y & & & \\ & Y_{Lx} & & \\ & & Z_{Tx} & \\ & & & U_z \end{bmatrix} \begin{bmatrix} I_y + i_y \\ e_{Lx} \\ i_{Tx} \\ E_z + e_z \end{bmatrix} = \begin{bmatrix} Y_y & & & \\ & Y_0 Z_{Lx} & & \\ & & Z_0 Y_{Tx} & \\ & & & Z_z \end{bmatrix} \begin{bmatrix} E_y + e_y \\ i_{Lx} \\ e_{Tx} \\ I_z + i_z \end{bmatrix} \quad (37)$$

Here the subscripts Tx and Lx refers to x-branches that turn out to be tree-branches and links, respectively.

Kirchhoff's voltage law in the form $e = De_T$ (see Equation (33) of reference 13) may then be expressed as

$$\begin{bmatrix} e_{Ty} \\ e_{Tx} \\ e_{Tz} \\ e_{Ly} \\ e_{Lx} \\ e_{Lz} \end{bmatrix} = \begin{bmatrix} U_{Ty} & 0 & 0 \\ 0 & U_{Tx} & 0 \\ 0 & 0 & U_{Tz} \\ -C_{Ty}^t & 0 & 0 \\ -C_{Tx}^t & -C_{Tz}^t & 0 \\ -C_{Ty}^t & -C_{Tx}^t & -C_{Tz}^t \end{bmatrix} \begin{bmatrix} e_{Ty} \\ e_{Tx} \\ e_{Tz} \end{bmatrix} \quad (38)$$

Similarly, Kirchhoff's current $i = C i_L$ may be written as

$$\begin{bmatrix} i_{Ty} \\ i_{Tx} \\ i_{Tz} \\ i_{Ly} \\ i_{Lx} \\ i_{Lz} \end{bmatrix} = \begin{bmatrix} C_{Ty} & C_{Tx} & C_{Tz} \\ 0 & C_{Tx} & C_{Tz} \\ 0 & 0 & C_{Tz} \\ U_{Ly} & 0 & 0 \\ 0 & U_{Lx} & 0 \\ 0 & 0 & U_{Lz} \end{bmatrix} \begin{bmatrix} i_{Ly} \\ i_{Lx} \\ i_{Lz} \end{bmatrix} \quad (39)$$

Next, we define D_y and C_z by the expressions (see Equations (32) and (34) of reference 13)

$$e_y = \begin{bmatrix} e_{Ty} \\ e_{Ly} \end{bmatrix} = \begin{bmatrix} U_{Ty} \\ -C_{Ty}^t \end{bmatrix} e_{Ty} = D_y e_{Ty} \quad (40)$$

and

$$i_z = \begin{bmatrix} i_{Tz} \\ i_{Lz} \end{bmatrix} = \begin{bmatrix} C_{Tz} \\ U_{Lz} \end{bmatrix} i_{Lz} = C_z i_{Lz} \quad (41)$$

Then, we invoke the cognate forms of Kirchhoff's laws $D^t i = 0$ and $C^t e = 0$. From the first of these, using only the first row of D^t as obtained from Equation (38), we find that

$$D_{Ty}^t i_y = C_{Tyx} i_{Lx} + C_{Tyz} i_{Lz} \quad (42)$$

and from the second, using only the last row of C^t as obtained from Equation (39), we have

$$C_z^t e_z = -C_{Ty}^t e_{Ty} - C_{Tx}^t e_{Tx} \quad (43)$$

Accordingly, if we premultiply the first row of Equation (37) by D_y^t and the last row by C_z^t , then eliminate e_y and i_z by using Equations (40) and (41), and rearrange terms, we find that

$$\begin{bmatrix} D_Y^t(I_Y - Y_Y E_Y) \\ 0 \\ 0 \\ C_Z^t(E_Z - Z_Z I_Z) \end{bmatrix} = \begin{bmatrix} D_Y^t Y_Y D_Y & 0 \\ Y_O Z_{Lx} & \\ 0 & Z_O Y_{Tx} \\ 0 & C_Z^t Z_Z C_Z \end{bmatrix} \begin{bmatrix} e_{Ty} \\ i_{Lx} \\ e_{Tx} \\ i_{Lz} \end{bmatrix} - \begin{bmatrix} D_Y^t i_Y \\ Y_{Lx} e_{Lx} \\ Z_{Tx} i_{Tx} \\ C_Z^t e_z \end{bmatrix} \quad (44)$$

Finally, extracting from Equations (38) and (39) the relations

$$e_{Lx} = -C_{Tyx}^t e_{Ty} - C_{Txx}^t e_{Tx} \quad (45)$$

and

$$i_{Tx} = C_{Txx} i_{Lx} + C_{Txz} i_{Lz} \quad (46)$$

we substitute these into Equation (44) along with Equations (42) and (43), obtaining the result

$$\begin{bmatrix} D_Y^t(I_Y - Y_Y E_Y) \\ 0 \\ 0 \\ C_Z^t(E_Z - Z_Z I_Z) \end{bmatrix} = \begin{bmatrix} D_Y^t Y_Y D_Y & -C_{Tyx} & 0 & -C_{Tyz} \\ Y_{Lx} C_{Tyx}^t & Y_O Z_{Lx} & Y_{Lx} C_{Txx}^t & 0 \\ 0 & -Z_{Tx} C_{Txx} & Z_O Y_{Tx} & -Z_{Tx} C_{Txz} \\ C_{Tyx}^t & 0 & C_{Txz}^t & C_Z^t Z_Z C_Z \end{bmatrix} \begin{bmatrix} e_{Ty} \\ i_{Lx} \\ e_{Tx} \\ i_{Lz} \end{bmatrix} \quad (47)$$

This equation includes not only the differential equations of the transmission line(s) but also both the algebraic and differential equations of the terminating networks. For the matrix Y_Y may consist of capacitance terms (Cd/dt) or conductance terms (G) or both while the matrix Z_Z may consist of inductance terms (Ld/dt) or resistance terms (R) or both. Explicit display of the algebraic equations separated from the differential equations can be effected by a straightforward extension of the procedure just described.

SUMMARY AND CONCLUSIONS

1. The method of characteristics has been applied to the problem of transient analysis of a lossy transmission line. An analytic solution has been obtained at a finite number of points along the line and at fixed time increments.
2. A general method of formulation has been developed which permits the input and output of a transmission line to be represented by simple equivalent networks. The equations of these equivalent networks are then combined with the equations for any arbitrary terminating networks that may be connected to the transmission line.

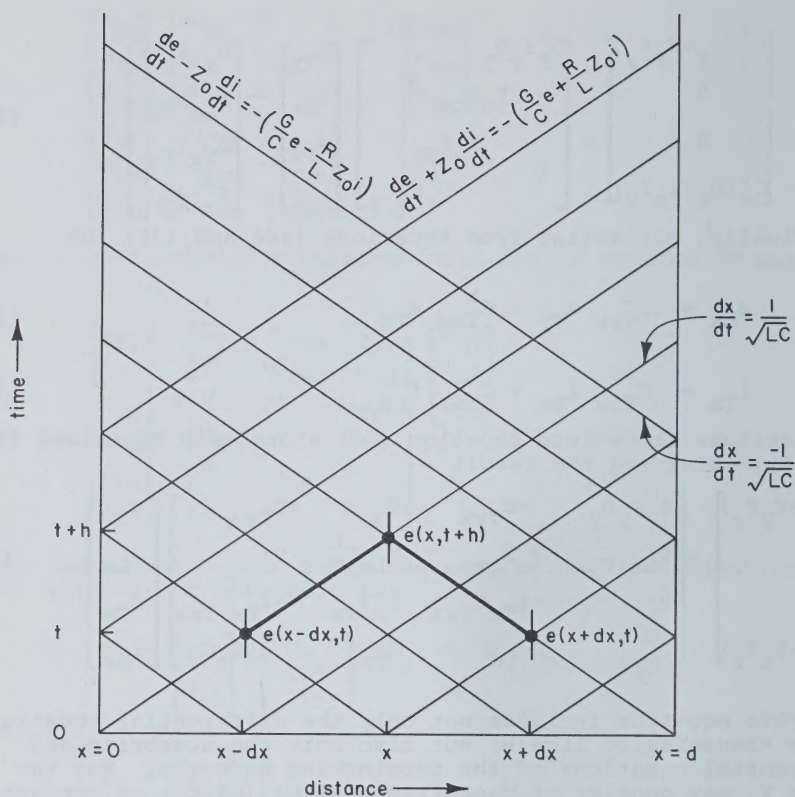


Figure 1 Characteristic Curves

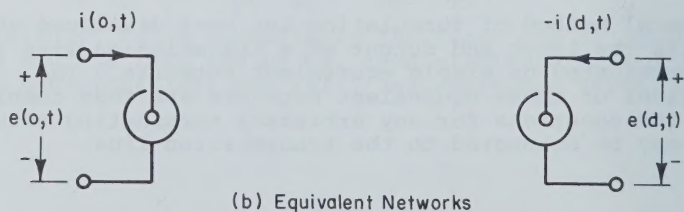
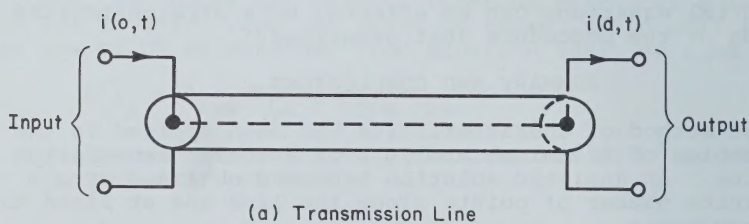


Figure 2 Transmission Line and Equivalent Networks

REFERENCES

- (1) F. H. Branin, Jr., "Transient Analysis of Lossless Transmission Lines", Proc. IEEE, 55, 2012-2013 (1967).
- (2) C. R. Pendred, "Transient Analysis of Transmission Lines with Nonlinear Terminations", Proc. IEEE, 52, 190 (1964).
- (3) A. G. Franco and A. S. Farber, "Transient Analysis of Transmission Line Tunnel Diode Circuits", Proc. IEEE, 51, 368-369 (1963).
- (4) W. Frey and P. Althammer, "The Calculation of Electromagnetic Transients on Lines by Means of a Digital Computer", The Brown Boveri Review, 48, (5/6) 344-355 (1961).
- (5) P. L. Arlett and R. Murray-Shelley, "An Improved Method for the Calculation of Transients on Transmission Lines Using a Digital Computer", Proc. P.I.C.A. Conference, 195-211 (1965). Sponsored by IEEE Power Group.
- (6) O. Schnyder, "Druckstosse in Pumpensteigleitungen", Schweizer Bauzeitungen, 94, 22-23 (1929).
- (7) L. Bergeron, Du coup de belier en hydraulique et coup de foudre en electricite, Ed. Dunod, Paris (1949). Published in English as: Water Hammer in Hydraulics and Wave Surges in Electricity, John Wiley and Sons, New York (1961).
- (8) H. W. Dammel, "A Method for Solving Transient Phenomena in Multiphase Systems", Proc. 2nd Power Systems Computation Conference, Stockholm, Sweden, (June 1966).
- (9) Y. K. Liu, "Transient Analysis of TEM Transmission Lines", Proc. IEEE, 56, 1090-1092 (1968).
- (10) F. H. Branin, Jr., "Transient Analysis of Lossless Transmission Lines", IBM Systems Development Division, Poughkeepsie, New York, Technical Report TR00.1615, 7/7/67.
- (11) R. Courant and K. O. Friedrichs, Supersonic Flow and Shock Waves, Interscience Publishers, New York (1948).
- (12) Mary Lister, "The Numerical Solution of Hyperbolic Partial Differential Equations by the Method of Characteristics", Chapter 15 of Mathematical Methods for Digital Computers, (A. Ralston and H. S. Wilf, editors), John Wiley and Sons, New York (1960).
- (13) F. H. Branin, Jr., "Computer Methods of Network Analysis", Proc. IEEE, 55, 1787-1801 (1967).

DISTRIBUTED RC NETWORK SYNTHESIS - WHAT IMPORTANT CHALLENGES REMAIN?

RALPH W. WYNDRUM, JR.

Bell Telephone Laboratories, Incorporated
Murray Hill, New Jersey

Today I want to stand back, if I may, to place into perspective the work that has been done in recent years on the broad and fascinating area of network synthesis accomplished by distributed networks. My comments will really be restricted to distributed RC ($\overline{RC}$) networks, since I intend to avoid a discussion of lossless (microwave) networks. My vantage point will be that of a person involved in integrated circuit design, who has also spent considerable time developing and implementing RC network synthesis procedures.

The development of distributed RC network synthesis enjoys some interesting parallels with the early development of lumped RC network synthesis. As we know the most significant practical thrust for lumped RC network synthesis came from the introduction of the transistor which made the inductor incompatible with economical and small active-network realizations. Passive RC networks had been vigorously developed for more than a quarter of a century at the time the transistor was introduced. Distributed RC networks, on the other hand, came into their own when it became apparent that the thin film and semiconductor integrated circuit technology made it feasible and desirable to realize inexpensive and precise distributed RC networks. And so, by 1961, Hager,⁽¹⁾ Happ⁽²⁾ and Castro⁽³⁾ among others, had identified the salient characteristics of distributed RC networks and tabulated significant properties of the basic structures. Now seven years later, it seem appropriate to ask what progress has been made, to identify specific trends and understandings that have come out as a result of the work that has been done in this area and to ask where we should go from here.

PASSIVE DISTRIBUTED RC SYNTHESIS PROCEDURES

Soon after the early analytical work on distributed RC networks was published various workers in the field immediately saw their way clear to some well ordered synthesis procedures. These are obviously closer to application than analysis procedures. The synthesis procedures often took into account eminently practical considerations. Some of the techniques, oriented to achieving truly integrated distributed RC networks, included the practical restriction of achieving distributed RC network functions with a common ground plane and continuous resistive and dielectric layers. Several of the synthesis procedures, as we shall see shortly, recognized the availability and indeed necessity of using the computer to effectively deal with the mathematics of distributed networks. The synthesis techniques quite properly, tried to recognize and relate the associated aspects of the fabrication and computational technologies. In this sense, they were far from being purely academic.

My first vugraph (Fig. 1) tries to classify some of the more significant synthesis procedures into just four groups. The first group (4-12,36) includes the work of Wyndrum, O'Shea, Scanlon, Rhodes, Stein, Lee, and much earlier Ozaki and Ishii. These approaches generally are based on using commensurate lengths of line and synthesizing a required function in the transformed frequency plane. Although the early work allowed for only URC segments, recent work has included broad classes of tapered networks.(37,38) Reference (38) has extended the early work to most tapered lines with closed form solutions. In general, the transformations are hyperbolic, sometimes

positive real, and succeed in representing distributed RC networks in the transformed plane by ratios of polynomials in the transformed variable. The synthesis procedure is carried out in the transformed plane where the mechanics of it become far simpler.

The second group, (13-16,⁴²) includes those approaches which attempt to simplify the synthesis procedure, possibly at the expense of the network fabrication feasibility and indeed practicality. Heizer, Fu and others, have shown that if it is possible to shape the geometry of the prescribed RC network, and possibly provide lossy dielectric layer with a prescribed loss function, we may assure ourselves that at least two of the "y" parameters are rational functions of the original complex frequency variable. This may be of real assistance in constructing rational transfer functions, but it would be my judgment that it is probably not necessary to limit ourselves in this fashion.

A third approach has been taken to broaden the class of network functions which may result from a prescribed synthesis technique. This approach is able to take into account networks which contain both distributed and lumped parameter elements and networks which contain incommensurate lengths of distributed transmission line. Early results which provided synthesis procedures did so only for certain specific cases: (1) incommensurate lines terminated in a resistor or lumped element load;^(19,20) (2) a variety of series and shunt elements with incommensurate unit elements. However, recent papers by Koga and Rao^(23,24) have gone far to solve the general problem of the synthesis of positive real functions and matrices of several variables. The related approximation problem though, for lumped-distributed RC networks, has been virtually neglected.

A substantially different approach to the theory of nonuniform $\overline{RC}$ lines has been taken by Protonotarios and Wing.^(26,27) They have studied the analytical properties of nonuniform $\overline{RC}$ lines in both the frequency and time domain. In connection with the frequency domain realization they have obtained necessary and sufficient conditions for an arbitrary function to be realizable as a network function of a general nonuniform distributed RC line. This line may be terminated in or contain arbitrary lumped RC networks. This study considers bounds on the network singularities of an arbitrary RC network which is characterized by a consistent set of ABCD parameters. The authors indicate in their conclusion that it is generally true that the frequency response of an arbitrary distributed network is quite insensitive to small variations in the per unit length resistance and per unit length capacitance of the network (i.e., "taper"). This can really be tied back to the fact that no matter what the taper of the network, the poles of a passive distributed RC network are all on the negative real axis and differ only slightly in their locations. Thus, variations in the overall transfer functions of these networks arise primarily from the transmission zeros particularly when the zeros are placed close to the $j\omega$ axis.

On a different topic, it should be noted that the synthesis and in fact, even the analysis of multilayer distributed RC networks represents an area of current concern. There is considerable work being pursued at present in this area, and to the extent that it provides a great deal of network function in a more limited area due to their use of multiple layers, they may have promise for future integrated circuit configurations. Nonetheless, it has to be noted that, at present, it is often not technically or economically feasible to lay down multiple layer structures with large numbers of layers.

Certain special structures, such as notch or rejection networks, have been studied extensively,⁽¹⁻³⁾ but they are usually lumped-distributed networks and have not been incorporated into formal synthesis procedures. They represent an important ingredient in active RC synthesis.

A common denominator of each of the previously mentioned approaches is the extent to which computer-aided design is intimately woven in with the synthesis. In fact, the computer aspects of the synthesis of the distributed RC networks came in at two points.

The first point is the approximation of the desired network transfer or driving point function. Let me stress that the approximation problem is a critical ingredient of any meaningful synthesis procedure, and all too often it is neglected. The second point is that the algebraic complexities of distributed networks, whether in the time domain or the frequency domain, demand the intelligent application of the modern computer. Even if the approach of transforming uniform distributed RC network hyperbolic functions to rational functions of a new frequency variable is taken, the computer becomes an invaluable tool for the repeated application, for example, of R chard's Theorem. To illustrate this, my second vugraph (Fig. 2) vividly indicates the contribution the computer can be expected to make in the design of integrated distributed RC networks. Once the synthesis procedure is chosen, the computer can implement it from the original graphical description of the prescribed transmission or driving point characteristics through the geometric mask design. This is an exciting state of affairs - it means that the computer can essentially replace the network theorist once the theorist has laid down an appropriate set of synthesis rules. Of course, we all know that the network theorist is not signing his own death warrant here - he must proceed to more exciting and more sophisticated challenges.

ACTIVE DISTRIBUTED RC NETWORKS

What, then, are the challenges which stand to be conquered over the next few years? I think if we expect distributed RC networks to find their place in practical integrated circuits they must clearly be augmented by and considered with active elements. In fact, I would submit that active network building blocks may have more flexibility and be better suited to integrated circuit realization when built with distributed, not lumped, RC networks. Smaller size, fewer interconnections and simpler photolithography help to give the advantage to distributed networks.

Let me describe to you, to begin with, a building block approach, seen in vugraph 3 (Fig. 3), that has been evolved by Chen and Levine⁽²⁸⁾ and augmented by Spaulding.⁽²⁹⁾ The structure in question is essentially a transversal filter for which the distributed RC line provides a dispersive medium. We see a uniform distributed RC line with a single input and with taps coming off the resistive path at a prescribed set of intervals, or tap spacings, along the line. The signals tapped off at each point ("transversal" to the direction of signal flow down the line) are summed up into either the positive or negative inputs of a differential amplifier stage. Chen and Levine have shown that it is possible to realize any finite energy function with this kind of filter (for example, bandpass, lowpass or high-pass filters) with a finite number of taps. This synthesis procedure was carried out originally in the time domain and to the extent that it is an approximation in the time domain, it is difficult to translate a specification directly to the frequency domain approximation. However, with uniform spacing, the structure falls into the class of a cascade of commensurate line segments, and as such may be treated mathematically by a suitable hyperbolic frequency transformation as was just described for passive networks.

The active element, in this case the differential amplifier, provides gain as well as phase inversion; the latter is critical to the synthesis procedure for it permits complex frequency zeros of transmission. We have gained enough experience with synthesizing required functions with this kind of structure to realize that a parallel exists between digital filters and the transversal filter in terms of their required complexity and effectiveness. In particular, both nonrecursive digital filters and the nonrecursive RC transversal filter structure which I have just described are inefficient in realizing a wide variety of network functions; they require far too many taps. In the transversal filter, we can see that the synthesis procedure just described really provides only for negative real poles but finite complex transmission zeros. The latter arises because the output function is formed from the differences of several tap output functions. It is thus a synthesis procedure which relies on the appropriate placement of a sufficient number of zeros for significant frequency selective characteristics. Yet the concept, and the rigorous analysis which Chen and Levine developed, is an important step forward in RC network synthesis.

As an extension to this approach it is of course, clear that a recursive structure should be more efficient since it should, and in fact, does permit complex poles to be achieved. This approach, shown in vugraph 4 (Fig. 4), has been taken by D. A. Spaulding,⁽²⁹⁾ among others. He has shown that just a few taps (perhaps four or five) is sufficient to realize quite sophisticated and selective filter characteristics. Spaulding bases his synthesis procedure on the theory of dominant poles and as we see in vugraph 5 (Fig. 5) convincingly demonstrates that for reasonably high Q complex poles this theory is sufficiently accurate to provide a solution to the approximation problem, akin to that of lumped networks. Further work in this area appears profitable, since Spaulding showed that the sensitivity functions associated with the open circuit voltage transfer function are quite reasonable, of the order of one. An extension which would provide for a building block with a variety of available taps for use in integrated circuit realizations over a wide range of frequencies would be a natural outgrowth of this procedure. With just a single tap, the structure reduces to a simple and previously reported building block.^(30,31)

Other active synthesis approaches have been suggested recently; for example, that of Parkin⁽³¹⁾ using NIC's and lumped-distributed networks (vugraph 6, Fig. 6). Although Parkin did not use the multivariable frequency approach described above for lumped-distributed passive networks, this approach might provide an orderly insight into such a synthesis procedure.

Fu and Fu⁽³⁴⁾ have indicated the use of NIC's and partitioned RC networks, but more investigation is called for here before a judgement of the practical applicability of this approach can be weighed.

Some studies have been carried out relative to the active distributed RC stability problem, and of course, the related questions of network sensitivity. It is not clear, however, that the classical lumped element sensitivity functions apply here. We must be concerned with the nonuniformity of the capacitance and resistance density functions, and then their deviation as a function of spacial coordinaters, which is indeed a new dimension of the problem.

The stability question is a tricky problem mathematically when distributed networks are involved; it should be approached with a great deal of caution.⁽³³⁾ It is not a new problem, and has been treated for other distributed mechanical and thermal systems.⁽³⁹⁻⁴¹⁾ If, for example, a transformed

frequency plane is used to carry out the design, there should be an accompanying knowledge of stable and unstable pole positions in the new plane. In the W-plane synthesis procedure, for example, as was described in the second vugraph, (where $W = \tanh a\sqrt{s/2}$) we may indicate the region of stable poles only after a careful analysis of the multiple-valued functions which are involved. This has been rigorously developed in connection with a recent NIC-RC building block synthesis procedure.⁽³⁵⁾ Vugraph 7 (Fig. 7) indicates the appropriate region of stability, together with an analytic expression for the boundary. Vugraph 8 (Fig. 8) shows lines of constant Q for dominant poles mapped from the original s-plane into the W-plane. These design curves are useful in carrying out the synthesis procedure within the W-plane, for the exponential polynomials identified in the second vugraph are not well behaved mathematical functions when the dominant poles are near the real frequency contour. A computer, however, can readily be called upon to iterate the desired transfer function until it is approximated with an arbitrary degree of closeness. A typical bandpass structure is seen in vugraph 9 (Fig. 9). A practical restriction on distributed RC network applications that we must observe is the degree to which R and C , and more importantly, appropriate RC products, can be held in manufacture, independent of temperature and other environmental conditions; this restriction will probably limit the practical application of distributed RC filters to the audio region or, depending on the severity of application requirements, to less than 100 KHz with the present state of applicable technology. Alternatively, it may be advantageous to incorporate low pass distributed RC networks into switched filter structures, as we do with lumped networks, so as to shift the burden of frequency stability from the passive components to a single carefully controlled oscillator.

CONCLUSION

I think, then, that a clear picture of important areas of future interest has evolved. Modern electronics means integrated electronics - for the dominant reasons of low cost, better reliability and smaller size. To the extent that distributed RC networks can contribute to the thrust of integrated circuits, distributed networks achieve importance in a practical sense.

Meaningful and widely used filters must provide useful frequency discrimination - characteristic of structures with complex poles - which passive networks, RC or $\overline{RC}$, cannot achieve. Passive distributed RC networks must be coupled, in a single hybrid integrated circuit package, with active elements to achieve acceptance and application. Today, linear amplifiers are inexpensive, and we can afford to include many transistors within silicon integrated circuits without degrading the cost of an active filter. In time, we may ask if we can afford the $\overline{RC}$ network.

Our distributed RC network synthesis procedures must take into account the environmental performance of the composite integrated network. Temperature compensation must be effected between the active and passive portions of the circuit. We must take advantage of the fact that batch fabricated components have nearly identical electrical properties and track with temperature. A different set of sensitivity parameters, which includes known, related interactions of various components, must be dealt with.

It is clear, then, that the use of distributed RC networks is inextricably tied to the active integrated circuit technology, since there is limited need for purely passive $\overline{RC}$ networks. We have made remarkable headway over synthesizing passive distributed RC networks. Continued rapid advance

relies less on the further analysis and synthesis of passive $\overline{RC}$ structures, and more on the appropriate inclusion of these structures in active networks. Computer aided design must be made to play a very important role - it is involved in all stages from the original functional approximation to the ultimate topological realization. Most critically, this realization must be molded to provide more economical, smaller, and perhaps more reliable building blocks - for the integrated-circuit active-filters of a tomorrow that is almost here today.

REFERENCES

1. C. K. Hager, "Distributed-Parameter Networks for Circuit Miniaturization," Proc. Electron. Components Conf., pp. 195-203, 1959.
2. W. Happ, "Synthesis of Distributed-Parameters RC Networks," Proc. IRE, Vol. 50, pp. 483-484, 1962.
3. P. S. Castro, "Microsystem Circuit Analysis," Elec. Eng., Vol. 80, pp. 535-542, 1961.
4. R. Wyndrum, Jr., "The Exact Synthesis of Distributed RC Networks," Tech. Rept. 400-76, New York University, May 1963. Also, 1965 IEEE Convention Record, Part 7, pp. 93-97.
5. R. P. O'Shea, "Synthesis Using Distributed RC Networks," IEEE Trans. Circuit Theory, Vol. CT-12, pp. 546-554, December 1965; also in, IEEE Internat'l Conv. Record, Vol. 13, Pt. 7, pp. 19-29, 1965.
6. J. O. Scanlon and J. D. Rhodes, "Cascade Synthesis of Distributed Networks," Symp. Proc. Polytech. Inst. (Brooklyn), Vol. XVI, pp. 227-256, 1966.
7. R. W. Newcomb, "Nonreciprocal Transmission Line n-port Synthesis," Proc. Inst. Radio Electron. Engr. (Australia), Vol. 26, pp. 135-142, April 1965.
8. J. J. Stein, J. H. Mulligan, Jr., and S. S. Shamis, "Realization of Real Transmission Zeros Using Uniform Distributed RC Networks with Common Ground Connections," IEEE Internat'l Conv. Record, Vol. 15, Pt. 6, pp. 98-105, 1967.
9. J. J. Stein, J. H. Mulligan, Jr., and S. S. Shamis, "Realization of Transfer Functions Using Uniform Distributed RC Networks with Common Ground Connections," Symposium on Modern Networks, (Proc.) Brooklyn Polytechnic Inst., Vol. XVI, pp. 149-172, 1966.
10. Henry Ronald Chinn, "The Synthesis of Exponentially Tapered $\overline{RC}$ Networks, Realizing Driving Point Immittance Functions," PhD. Thesis, U. of British Columbia, Dec. 1966.
11. T. N. Rao, C. V. Shaffer, and R. W. Newcomb, "Realizability Conditions for Distributed RC Networks," ERL Tech. Rept. 6558-7, Stanford University, May 1965.
12. P. E. Bassett, "Cascade Networks of Symmetrical Proportioned Elements," J. Franklin Inst., Vol. 285, pp. 222-228, March 1968.

13. K. W. Heizer, "Distributed RC Networks with Rational Transfer Functions," IRE Trans. Circuit Theory, Vol. CT-9, No. 3, pp. 356-362, 1962.
14. K. W. Heizer, "Rational Parameters with Distributed Networks," IEEE Trans. Circuit Theory, Vol. CT-10, No. 4, pp. 531-32, 1963.
15. B. B. Woo and R. G. Hove, "Synthesis of Rational Transfer Functions with Thin Film Distributed-Parameter RC Active Networks," Proc. Nat'l Electron. Conf., Vol. 21, pp. 241-246, 1965.
16. Y. Fu and J. S. Fu, "n-port Rectangularly Shaped Distributed RC Networks," IEEE Trans. Circuit Theory, Vol. CT-13, pp. 222-226, 1966.
17. Y. Oono and T. Koga, "Synthesis of a Variable Parameter One-Port," IEEE Trans. on Circuit Theory, Vol. 10, pp. 213-226, June 1963.
18. H. G. Ansell, "Networks of Transmission Lines and Lumped Reactances," IEEE Trans. Circuit Theory, Vol. CT-11, pp. 214-223, 1964.
19. M. Saito, "Synthesis of Transmission Line Networks by Multivariable Techniques," Symp. on Modern Networks, (Proc.) Brooklyn Polytech. Inst., Vol. XVI, pp. 353-392, 1966.
20. M. Saito, "The Condition for Unit Network Extraction in Mixed Lumped and Distributed Networks," IEEE Trans. Circuit Theory, Vol. CT-13, pp. 218-220, 1966.
21. D. C. Youla, "Synthesis of n-ports Containing Lumped and Distributed Elements," Symp. on Modern Networks, (Proc.) Brooklyn Polytech. Inst., Vol. XVI, 1966.
22. D. C. Youla, P. Tissi, and W. Kohler, "The Synthesis of Networks Containing Lumped and Distributed Elements - Parts I and II," Techn. Rept. RADG-TR-66-489, Polytechnic Inst. of Brooklyn, 1966.
23. T. Koga, "Synthesis of Finite Passive n-ports with Prescribed Positive Real Matrices of Several Variables," IEEE Trans. on Circuit Theory, Vol. CT-15, No. 1, March 1968, pp. 2-23.
24. T. N. Rao and R. W. Newcomb, "Synthesis of Lumped-Distributed RC n-ports," IEEE Trans. Circuit Theory, Vol. CT-13, pp. 458-459, 1966.
25. T. N. Rao, "Synthesis of Lumped-Distributed RC Networks," Stanford Univ. Dept. of Electrical Engineering PhD. dissertation, May 1967.
26. E. Protonotarios and O. Wing, "The Distribution of Zeros of ABCE Parameters of Arbitrary RC Transmission Lines," Proc. Allerton Conf., pp. 162-171, 1965.
27. E. Protonotarios and O. Wing, "Theory of Nonuniform RC Lines, Part I - Analytic Properties and Realizability Conditions in the Frequency Domain, Part II - Analytic Properties in the Time Domain," IEEE Trans. Circuit Theory, Vol. CT-14, pp. 2-20, March 1967.
28. W. I. H. Chen and R. C. Levine, "Computer-Designed Transversal Filters Using Thin Film RC Dispersion," Proc. 10th Midwest Symp. Circuit Theory, Purdue University, Lafayette, Ind., 1967.

29. D. A. Spaulding, "Synthesis of Rational Transfer Function Approximations Using a Tapped Distributed RC Line with Feedback," to be published in B.S.T.J.
30. L. P. Huelsman and W. J. Kerwin, "Digital Computer Analysis of Distributed Lumped Active Networks," IEEE Journal of Solid-State Circuits, V. SC-3, No. 1, pp. 26-29, March 1968.
31. R. W. Wyndrum, Jr., "Active Distributed RC Networks," IEEE Journal of Solid-State Circuits, V. SC-3, No. 2, September, 1968.
32. R. E. Parkin, "The Design of Active Distributed Filters," B.S.T.J., V. 47, No. 5, pp. 605-622, May-June 1968.
33. J. J. Bourquin and T. N. Trick, "Stability of Distributed RC Networks," Proc. 10th Midwest Symp. Circuit Theory, Purdue University, Lafayette, Ind., 1967.
34. Y. Fu and J. Fu, "Synthesis of Active Distributed RC Networks," IEEE Trans. Circuit Theory, V. 13, No. 2, pp. 259-264, September 1966.
35. R. W. Wyndrum, Jr., R. Barcklow and W. Woronka, "Active $\overline{RC}$ Filters Using Negative Impedance Converters," to be published.
36. S. C. Lee and S. Penbeci, "A General Approximation Synthesis Method Using Uniform RC and RCG Lines," Proc. Fifth Annual Allerton Conf., 1967.
37. S. C. Lee and S. Penbeci, "Synthesis of Distributed Networks by Means of Orthogonal Functions," PGCT Transactions, to be published February 1969.
38. S. C. Lee, "Synthesis of Tapered RCG Networks," PGCT Transactions, to be published February 1969.
39. W. H. Howe, "Recent Developments in Process Control," Chapter of Automatic and Manual Control, A Tustin, ed., N.Y. Academic Press, 1952.
40. Y. Takahashi, "Transfer Function Analysis of Heat Exchange Processes," ibid.
41. D. A. Pierre and T. J. Higgins, "Pole Location and Stability of Distributed Parameter Feedback Systems," Proc. NEC, Vo. 20, pp. 801-803, 1964.
42. S. Y. Hwang, "RC Networks with Rational Y-Parameters Having Prescribed Poles," PGCT Transactions (to be published).

FIG. 1
PASSIVE $\overline{RC}$ NETWORK SYNTHESIS
PROCEDURES

USING FREQUENCY TRANSFORMATIONS OF $\overline{RC}$ TO L,C OR R,C	USING PARTITIONED ELECTRODES FOR RATIONAL FUNCTIONS
WYNDRUM (1963)	HEIZER (1962,3)
O'SHEA (1965)	WOO/HOVE (1965)
SCANLON/RHODES (1965)	FU/FU (1966)
NEWCOMB (1965)	
STEIN/MULLIGAN/SHAMIS (1966)	
CHINN (1966)	
RAO/SHAFFER/NEWCOMBE (1966)	
BASSETT (1968)	
USING MULTIVARIABLE (LUMPED / DISTRIBUTED) FUNCTIONS	OTHER RESULTS GENERAL REALIZABILITY OF NON-UNIFORM $\overline{RC}$ LINES
OONO/KOGA (1963)	PROTONOTARIOS /WING (1967)
ANSELL (1964)	MULTILAYER $\overline{RC}$ NETWORKS
SAITO (1966-7)	YANAI (1963)
YOU LA (1966)	BASSETT (1966)
KOGA (1967)	CHANG /WING (1967)
RAO (1967)	

FIG. 2
COMPUTER GENERATION OF $\overline{RC}$ NETWORKS

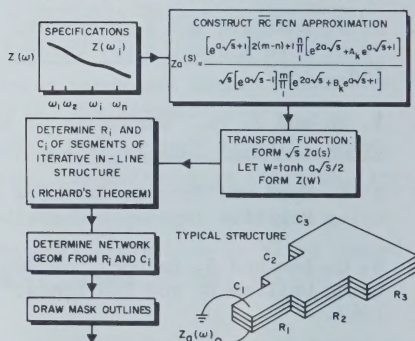


FIG. 3
TAPPED $\overline{RC}$ LOW PASS NETWORK

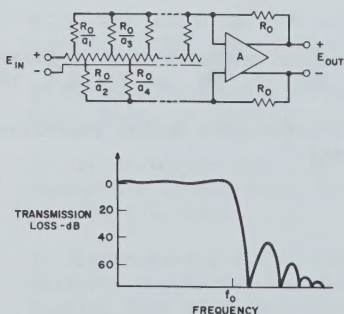


FIG. 4

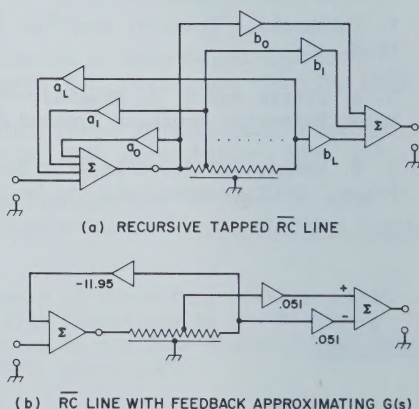


FIG. 5

GAIN VS FREQUENCY FOR G (s)

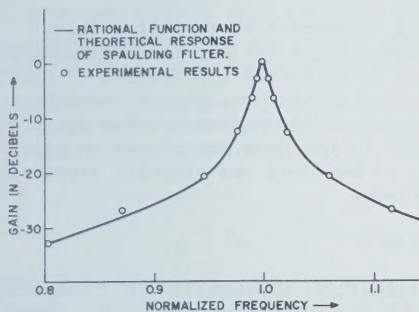


FIG. 6

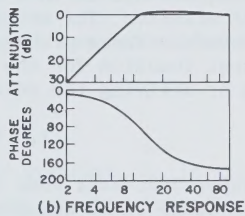
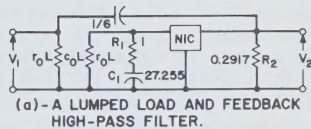


FIG. 7

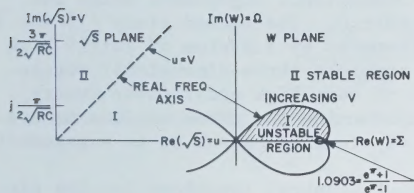
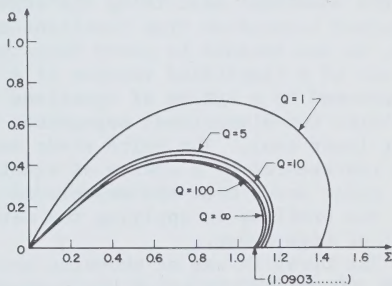
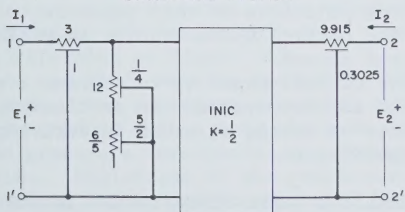
 $\sqrt{s} \rightarrow W$ PLANE MAPPING

FIG. 8

LINES OF CONSTANT Q MAPPED INTO W PLANE

FIG. 9
BANDPASS FILTER

$$T = RC = 3$$

$$\Sigma = 1.10$$

VALUES IN OHMS, FARADS

THE APPLICATION OF THE METHOD OF STEEPEST ASCENT TO A VERY HIGH ORDER
OPTIMIZATION PROBLEM: THE PROBLEM OF DEVELOPING EVASION TACTICS FOR
AIRCRAFT PURSUED BY MISSILES

B.S.MORGAN, Jr., J.F.SCHAEFER, D.H.deDOES, M.D.CILETTI, G.COOK
Frank J. Seiler Research Laboratory
US Air Force Academy, Colorado 80840

ABSTRACT

The method of steepest ascent is well documented in the literature. However its application to problems of high order (over 20) is not straightforward. This paper discusses two problem areas in the application of the method of steepest ascent to a realistic model of the engagement between an aircraft and missile. The method is used to obtain optimum evasive tactics for the aircraft. These problem areas are: 1) how does one choose an appropriate control function for the aircraft; 2) how does one pinpoint errors in programming or deriving the adjoint equations.

I. INTRODUCTION

This paper discusses some of the difficulties encountered in a computer optimization study of an engagement between an aircraft and missile. The purpose of the study was the determination of optimum evasive tactics for the aircraft when attacked by a missile. The optimization criterion was the maximization of the distance of closest approach between the missile and the aircraft. The optimization technique was the method of steepest ascent [1].

Since the optimization criterion was different than that normally used and the equations describing the engagement were non-linear, time-varying, contained saturation type functions and functions available only as tabular data, it was decided to study three particular engagements. The first study [2] was of a simplified version of a two-dimensional engagement which was represented by a system of equations of order 5. The second study [3] was a realistic two-dimensional engagement represented by a system of fairly high order (over ten). The third study was a complete three-dimensional engagement represented by a system of equations of very high order (over twenty). This paper draws upon the experience of performing the three studies to discuss two problems in applying the method of steepest ascent to an engagement model of high order.

The first is one of choosing appropriate control functions for the aircraft. Normally the pilot has control over throttle setting, bank angle, and load factor. Selection of appropriate representations for thrust (throttle setting) and roll rate (subject to a two-sided inequality constraint) is straightforward. However, the allowable load factor depends upon either the maximum structural limit or the stall boundary depending upon the aircraft's mach number. Hence, the maximum load factor is state dependent. Therefore, one objective of this paper is to discuss a particular representation for the control which results in minimum complications in an application of steepest ascent.

The second objective of this paper is to present a systematic method of pinpointing errors in the adjoint system. In particular, for systems of high order the probability of making an error in deriving and/or coding the adjoint equations is significant.

II. THE GENERALIZED THREE-DIMENSIONAL AIRCRAFT MODEL

The aircraft was assumed to be a point mass whose translation in inertial space was based on the sums of acceleration due to thrust, lift, drag,

and gravity effects. External forces other than gravity were assumed to act along the flight path of the aircraft and normal to it in the direction of the vertical stabilizer [3]. Control of the aircraft is accomplished through its normal acceleration and roll rate (subject to airframe or pilot limits), and throttle setting (subject to engine limits). The purpose of the method of steepest ascent was thus to find the control vector (as a function of time) which yields the maximum miss distance for given initial conditions. In applying the method of steepest ascent a problem arose due to the tabular form given by much of the aircraft data such as drag in terms of lift coefficient (C_L) and mach number. Hence, it was not immediately apparent what should be the normal acceleration control for the aircraft. It could be the normal force, C_L , etc. The choice of the normal acceleration control is strongly influenced by two factors: the stall characteristics and the structural limit of the aircraft.

The normal acceleration, A_N , of the aircraft can be expressed

$$A_N = \frac{C_L S}{m} \frac{\rho V^2}{2}, \quad (1)$$

where S is wing surface area, m is the mass of aircraft, C_L is the coefficient of lift, ρ is density, and V the velocity of the aircraft. Because of the structural limit of the aircraft, the following constraint exists,

$$|A_N| \leq N_{\max} g, \quad (2)$$

where $N_{\max}$ is the maximum load factor of the aircraft, and g is the acceleration due to gravity. This can be written as

$$\frac{|C_L| \rho V^2 S}{2m} \leq N_{\max} g \quad (3)$$

or

$$|C_L| \leq \frac{2m N_{\max} g}{\rho V^2 S}. \quad (4)$$

Because stall may occur a second constraint exists,

$$|C_L| \leq C_{L_{\max}} (M_A), \quad (5)$$

where $C_{L_{\max}}$ is the maximum lift coefficient which is a function of the aircraft mach number, M_A .

Both constraints must be satisfied. Their relative importance depends on the aircraft velocity. At high velocities the structural constraint is most important and the major restriction. At low velocities the stall is most important and the major restriction. The velocity where the two constraints intersect is called the corner velocity. These constraints can be more clearly seen by referring to Figure (1).

In the literature one finds several techniques for dealing with control constraints of this type. One procedure penalizes constraint violations by augmenting the cost functional with a term which in some manner sums these violations. A second procedure introduces an additional state variable for each control constraint. Violations of the constraints thus contribute to missed terminal conditions which must be corrected by the method of steepest ascent. The primary objection to these techniques is the requirement for choosing an appropriate weighting factor in the former case and an increase in dimension of the state vector in the latter case.

The procedure ultimately selected places hard constraints on the load factor. That is, the control is set equal to the boundary value whenever the limit load factor or stall conditions are exceeded. Since the maximum lift coefficient is state dependent the adjoint system must be augmented with additional terms whenever C_L is set equal to $C_{L_{\max}}$. The mathematical manipulations required in this situation are discussed by Bryson [1].

In order to avoid testing for the two separate constraints it was decided to introduce an artificial control variable α and to take

$$C_L = \tanh(\alpha) [C_{L_{\max}}] \quad (6)$$

Since

$$|\tanh(\alpha)| \leq 1, \quad (7)$$

the stall constraint given by (5) is automatically satisfied. The structural constraint (3) can be written as

$$|\tanh(\alpha)| C_{L_{\max}} \leq \frac{2m N_{\max} g}{\rho V^2 S} \quad (8)$$

or

$$\tanh(|\alpha|) \leq \frac{2m N_{\max} g}{\rho V^2 S C_{L_{\max}}} \quad (9)$$

or

$$|\alpha| \leq \tanh^{-1} \left[\frac{2m N_{\max} g}{\rho V^2 S C_{L_{\max}}} \right] \quad (10)$$

Thus by using the artifice of the hyperbolic tangent it is necessary to check the constraint given by (10) since the stall constraint is automatically satisfied. Also, no staging occurs, i.e. the form of the control is the same at low and high velocities. The disadvantage is that the control constraint is state dependent which requires additional terms in the adjoint equations as was discussed earlier.

Other formulations for the control are possible and were considered but all were rejected as being less desirable. As an example, one formulation considered was

$$C_L = u C_{L_{\max}}, \quad |u| \leq 1, \quad (11)$$

at velocities below the corner velocity and

$$N g = u N_{\max} g, \quad |u| \leq 1, \quad (12)$$

at velocities above the corner velocity. The advantage to this formulation is that the control constraint is always state independent so additional terms in adjoints are not required. However, a disadvantage exists in that the normal acceleration enters the equations as

$$A_N = u \frac{C_{L_{\max}}}{2m} \rho V^2 S, \quad (13)$$

for velocities below the corner velocity, and as

$$A_N = u N_{\max} g \quad (14)$$

for velocities above the corner velocity. This "staging" causes jumps (discontinuities) in the adjoints which create considerably more problems than desired.

III. TOTAL VARIATION OF THE COST FUNCTIONAL

In this section we derive an expression for the total variation of the payoff function which is fundamental to the method of steepest ascent. For a complete mathematical description of steepest ascent see Bryson [1].

Let the system be represented by the set of first order differential equations with x a state vector and u the control vector,

$$\dot{x} = f(x, u, t). \quad (15)$$

Define a payoff functional in terms of the final state, $x(T)$ and final time, T , as:

$$\Phi = \Phi(x(T), T). \quad (16)$$

Such a formulation causes no loss in generality.

One now forms the incremental system equations

$$\delta \dot{x} = \left(\frac{\partial f}{\partial x} \right)_0 \delta x + \left(\frac{\partial f}{\partial u} \right)_0 \delta u \quad (17)$$

where the subscript zero indicates evaluation along a trajectory about which the perturbations are taken.

A set of adjoint equations are defined by

$$\dot{\lambda}_\Phi = - \left(\frac{\partial f}{\partial x} \right)_0^T \lambda_\Phi \quad (18)$$

where λ_Φ is the adjoint column vector and the superscript T means that the transpose of the matrix $\partial f / \partial x$ is to be taken.

By transposing (18) and post-multiplying it by δx one obtains

$$\lambda_\Phi^T \delta \dot{x} = - \lambda_\Phi^T \left(\frac{\partial f}{\partial x} \right)_0 \delta x. \quad (19)$$

Now pre-multiply (17) by λ_Φ^T to obtain

$$\lambda_\Phi^T \delta \dot{x} = \lambda_\Phi^T \left(\frac{\partial f}{\partial u} \right)_0 \delta u + \lambda_\Phi^T \left(\frac{\partial f}{\partial x} \right)_0 \delta x \quad (20)$$

Adding (19) and (20) yields

$$\frac{d}{dt} (\lambda_\Phi^T \delta x) = \lambda_\Phi^T \left(\frac{\partial f}{\partial u} \right)_0 \delta u, \quad (21)$$

which after integration becomes

$$\lambda_\Phi^T \delta x|_T = \int_{t_0}^T \lambda_\Phi^T \left(\frac{\partial f}{\partial u} \right)_0 \delta u \, dt + \lambda_\Phi^T \delta x|_{t_0}. \quad (22)$$

If one chooses

$$\lambda_\Phi^T(T) = \frac{\partial \Phi(x(T), T)}{\partial x(T)}, \quad (23)$$

then (22) can be written as

$$\delta \Phi(x(T), T) = \int_{t_0}^T \lambda_\Phi^T \left(\frac{\partial f}{\partial u} \right)_0 \delta u \, dt + \lambda_\Phi^T \delta x|_{t_0}. \quad (24)$$

We are interested in the total variation in the payoff function,

$$d\phi = \delta\phi(T) + \dot{\phi}(T) dT. \quad (25)$$

In view of (24) the total variation of ϕ can be written as:

$$d\phi = \int_{t_0}^T \lambda_{\phi}^T \left(\frac{\partial f}{\partial u} \right)_0 \delta u dt + \lambda_{\phi}^T \delta x \Big|_{t_0} + \dot{\phi} dT \quad (26)$$

IV. CHECKS ON THE ADJOINTS

The purpose of this section is to discuss procedures used to verify the adjoint equations. In particular, for systems of high order the probability of making an error in deriving and/or coding the $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial u}$ is significant.

In passing, it is noted that for the three-dimensional engagement between the aircraft and missile there were over 200 non-zero terms in the $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial u}$ matrices. Thus tests which will uncover such errors are important.

The first test uses the fundamental relation derived earlier

$$d/dt(\lambda_{\phi}^T \delta x) = \lambda_{\phi}^T \left(\frac{\partial f}{\partial u} \right)_0 \delta u. \quad (21)$$

It is clear that if one constrains

$$\delta u \equiv 0,$$

that the inner product $\lambda_{\phi}^T \delta x$ should be constant over the integration interval.

To verify this, two forward integrations of the system equations are made with slightly different initial conditions. The difference in the two resulting state histories is taken as δx . The adjoints are then integrated, after either of the two forward integrations is used as nominal to evaluate $\left(\frac{\partial f}{\partial x} \right)_0$. This test will be satisfied ($\lambda_{\phi}^T \delta x = \text{constant}$) if the equations have

been correctly derived and programmed. However, if the test fails, one knows that an error exists but may have considerable trouble pinpointing it because of coupling in the adjoint equations. Note that, as was our experience, the individual products in the $\lambda_{\phi}^T \delta x$ inner product typically varied by several orders of magnitude. Thus one must examine each $\lambda_{\phi_i} \delta x_i$ in order to determine the order to which the scalar product must agree over the interval $[t_0, t_f]$. It should be further noted that the inner product will not remain

constant if the forward and backward integration of the state and adjoint system are not accurate. In the three missile-aircraft engagement models considered it was found that the backward integration of the adjoints is very sensitive to step size if a fixed step integration procedure (such as a fourth-order Runge Kutta) is employed. Thus if the adjoint system has been derived and coded correctly one could possibly use the tests described in this section to gain insight as to the integration accuracy required in a particular problem.

The second test is used to validate individually each component of the adjoint vector. This is done by utilizing (26). Here, if δu is taken as zero and the integration is carried out to a fixed final time (thus $dT = 0$), then

$$d\phi = \lambda_{\phi}^T \delta x \Big|_{t_0} \quad (27)$$

This is implemented by first generating a nominal trajectory and calculating the payoff $\phi(x^*)$. One also determines a $x^*(t)$, from which $\frac{\partial f}{\partial x^*}$ can be

obtained and so $\lambda^*(t)$ can be computed from (18). The individual components x_i of the state x are perturbed one at a time such that

$$x_i|_{t_0} = x_i^*|_{t_0} + \delta x_i, \quad \delta x_j = 0 \quad j \neq i, \quad (28)$$

the x is integrated forward again yielding a new $\phi_i(x)$. One then determines $d\phi_i$ from

$$d\phi_i = \phi_i(x) - \phi(x^*). \quad (29)$$

Then if

$$d\phi_i \equiv \lambda_i^* \delta x_i|_{t_0} \quad (30)$$

all is well with that component of the adjoint vector.

The third test is used to find errors in the derivation and coding of the Jacobian $\partial f / \partial x$. The test proceeds as follows:

Step 1: First integrate nominal trajectory forward to some point t^* in time.

Step 2: Store $f^* = f(x^*, u^*, t^*)$ where $\dot{x} = f(x, u, t)$.

Step 3: Individually perturb the components of x^* . (let $x_i = x_i^* + \delta x_i$. We suggest $\delta x_i = .001 x_i^*$.)

Step 4: Evaluate $f = f(x^* + \delta x_i)$.

Step 5: Evaluate $a_{ij} = \left(\frac{\partial f_j}{\partial x_i} \right) \bigg|_{x=x^*} \approx \frac{f_j - f_j^*}{\delta x_i}$.

Step 6: Evaluate at t^* the Jacobian as furnished by the backward integration and compare.

V. ILLUSTRATIVE EXAMPLE

A) A Two-Dimensional Engagement Model

Presented in this section are some numerical results for checks on the adjoints in the simplified two-dimensional engagement [2]. The equations of motion for this system are:

$$\dot{R} = V_A \cos(\alpha - \theta) \quad (31)$$

$$\dot{\theta} = V_A / R \sin(\alpha - \theta) \quad (32)$$

$$\dot{r} = V_M \quad (33)$$

$$\dot{\xi} = (\theta - \xi) k \quad (34)$$

$$\dot{\alpha} = u \quad (35)$$

The state variables R , θ , r , ξ are defined in Figure 2. The time rate of change of the aircraft's flight path angle in the vertical plane is used as the control variable. To make the engagement realistic a constraint was placed on this rate of change. Hence,

$$|\dot{\alpha}| < 32.2 \frac{G_{\max}}{V_A} \quad (36)$$

where $G_{\max}$ is the aircraft's g limit. Note that in the above system of equations we have assumed that the missile stays close to the radar line of sight.

The payoff functional for the engagement is the distance of the closest approach between the aircraft and missile. Thus the aircraft attempts to maximize the following expression:

$$\phi = \min_t [(R-r)^2 + r^2(\theta - \xi)^2] . \quad (37)$$

The adjoint system corresponding to (31) through (35) is

$$\frac{d}{dt}\underline{\lambda} = [A]\underline{\lambda} \quad (38)$$

where

$$A = \begin{bmatrix} 0 & \frac{V_A}{R^2} \sin(\alpha - \theta) & 0 & 0 & 0 \\ -V_A \sin(\alpha - \theta) & \frac{V_A}{R} \cos(\alpha - \theta) & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ V_A \sin(\alpha - \theta) & -\frac{V_A}{R} \cos(\alpha - \theta) & 0 & 0 & 0 \end{bmatrix}$$

and where $\underline{\lambda}$ at the time of closest approach, (t_I) , is given by:

$$\lambda_1(t_I) = 2(R - r) \quad (39)$$

$$\lambda_2(t_I) = 2r^2(\theta - \xi) \quad (40)$$

$$\lambda_3(t_I) = -2(R - r) + 2r(\theta - \xi)^2 \quad (41)$$

$$\lambda_4(t_I) = -2r^2(\theta - \xi) \quad (42)$$

$$\lambda_5(t_I) = 0 \quad (43)$$

B) Numerical Results

Subsequent to programming the simple two-dimensional model the three tests described in Section III were applied with the following result:

Test 1		Test 2		
Integration Step No.	$\lambda^T \delta x$		dφ Actual	dφ Predicted
100	- 6724.677283	δR = 0.1	11187.763427	101696.228671
200	- 77811.999425	δθ = 0.0001	6706.777587	- 1396.973715
300	-226772.170650	δr = 0.1	-11659.280029	- 11659.282215
400	-293714.247284	δξ = 0.001	- 51.871582	- 51.872988
500	-204107.420398	δα = 0.0001	-66990.009033	14021.610141

TABLE 1

Test 3		
	Numerical Derivative	Partial Derivative Computed During Backward Integration
A ₁₂	0.00000235	0.04806219
A ₂₁	-980.39493084	-980.39023339
A ₂₂	0.00965933	0.00965071
A ₅₁	980.43291516	980.43023339
A ₅₂	- 0.00962560	- 0.00965071

TABLE 2

Test 1 indicates that the adjoints are incorrect since $\lambda^T \delta x \neq \text{constant}$ throughout the interval $[t_0, T]$. Test 2 shows that the trouble is due to either (or a combination of) $\lambda_1, \lambda_2, \lambda_5$. The results of Test 3 indicates that A₁₂ is probably incorrect. A subsequent check of the program revealed that R had not been squared in the denominator of A₁₂.

After performing the required correction to A₁₂ the three tests were again applied with the following result:

Test 1		Test 2		
Integration Step No.	$\lambda^T \delta x$		dφ Actual	dφ Predicted
48	62.150427	$\delta R = 0.1$	687.979614	687.977483
192	62.150486	$\delta \theta = 0.0001$	4791.365570	4791.608787
336	62.151765	$\delta r = 0.1$	- 727.303863	- 727.313687
480	62.157221	$\delta \xi = 0.001$	- 25.350921	- 25.351252
576	62.157143	$\delta \alpha = 0.0001$	-4788.102111	-4788.073662

TABLE 3

Test 3		
	Numerical Derivative	Partial Derivative Computed During Backward Integration
A ₁₂	0.00000250	0.00000250
A ₂₁	-999.98245864	-999.98320172
A ₂₂	0.00028637	0.00028981
A ₅₁	999.98304564	999.98320172
A ₅₂	- 0.00028368	- 0.00028981

TABLE 4

Note that for Test 1 the perturbation in the state vector, $\delta \mathbf{x}|_{t_0}$, is defined by the component perturbations (δR , δr , etc.) used in Test 2. Also note that the numerical results in Tables 3 and 4 are different than those in Tables 1 and 2. This is due to the fact that the engagement geometry was changed prior to obtaining the results in Tables 3 and 4.

VI. REFERENCES

1. Bryson, A. E., et al, "Optimal Programming Problems With Inequality Constraints": Part I, "Necessary Conditions for Extremal Solutions", AIAA Journal, Nov. 1963, p. 2544; Part II, "Solution by Steepest Ascent", AIAA Journal, Jan. 1964, p. 25.
2. Vachino, R. F., et al, "The Application of the Method of Steepest Descent to a Pursuit-Evasion Problem", AIAA Paper No. 68-878, AIAA Guidance, Control, and Flight Dynamics Conference, Pasadena, Calif., Aug. 1968.
3. deDoes, D. H., et al, "Computer Optimization Study of a Tactical Aircraft-Missile Engagement", Proceedings of the Rand Corporation Symposium on the Analysis and Design of Fighter Systems for Air-to-Air Combat, Santa Monica, Calif., Jan. 1968.

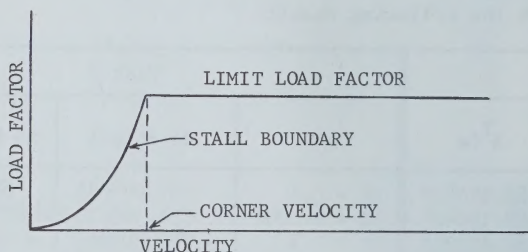


Figure 1 - Typical N vs. V Diagram

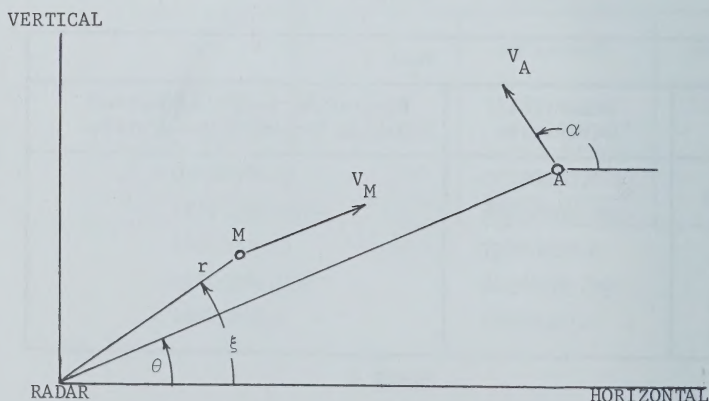


Figure 2 - Evasion Geometry

VACLAV DOLEZAL

State University of New York at Stony Brook, N. Y., and Mathematical Institute of the Czechoslovak Academy of Sciences

In the paper [1], J. Kudrewicz has introduced a new concept of stability - the so called energetic stability of a dynamical system. What he does is this: Assume that all signals and responses of a system belong to a set E defined as the set of all functions x on $[0, \infty)$ such that

$$v(x) = \lim_{T \rightarrow \infty} \left(\frac{1}{T} \int_0^T |x(t)|^2 dt \right)^{\frac{1}{2}} < \infty.$$

Let the signal s and the corresponding response r of a system be related by $r = Gs$, where G is an operator mapping E into itself; then the system under consideration is called energetically stable, if $v(Gs) = 0$ whenever $v(s) = 0$.

In the present paper the concept of energetic stability is modified and generalized for abstract spaces of signals, and three theorems on stability of feedback systems are given.

1. First, let us carry out some preliminary considerations.

Let Ω be a fixed set of real numbers such that $\sup \Omega = \infty$; if $T \in \Omega$, define $[T] = (-\infty, T] \cap \Omega$.

Further, let $\mathfrak{E}$ be a nonempty linear set, and let $\tilde{F}$ be the family of all mappings from Ω into $\mathfrak{E}$. With ordinary operations of addition and multiplication by a constant $\tilde{F}$ is a linear set.

Moreover, let F and F^* be nonempty linear subsets of $\tilde{F}$ such that F^* is a Banach space and $F^* \subset F \subset \tilde{F}$.

For each $T \in \Omega$ let us have a linear mapping S_T from $\tilde{F}$ into itself which satisfies the following conditions:

- (i) $S_{T_1} S_{T_2} = S_{T_1 + T_2}$ for any $T_1 \leq T_2$, $T_1, T_2 \in \Omega$.
- (ii) Let $x, y \in \tilde{F}$ and let $T \in \Omega$; then $x(t) = y(t)$ on $[T]$, iff $S_T x = S_T y$.
- (iii) Let $x \in \tilde{F}$; then $x \in F$ iff $S_T x \in F^*$ for all $T \in \Omega$.
- (iv) If $x \in F^*$, then $\|S_T x\| \leq \|x\|$ for any $T \in \Omega$.

We will also use the notation $S_T x = x_T$. Before proceeding further, let us present a few examples of specific sets $\tilde{F}$, F , F^* and mappings S_T which satisfy the above axioms (i) - (iv).

a) Let $\Omega = [0, \infty)$ and $\mathfrak{E} = R^n$ (n -dimensional Euclidean space with norm $|\cdot|$); then $\tilde{F}$ is the set of all n -vector valued functions on $[0, \infty)$. If $1 \leq p \leq \infty$ and we put

$$F = L_p = \{x: x \in \tilde{F}, x \text{ measurable}, \int_0^\tau |x(t)|^p dt < \infty \text{ for any } 0 < \tau < \infty\},$$

$$F^* = L_p = \{x: x \in F, \int_0^\infty |x(t)|^p dt < \infty\}$$

with the customary norm, then the mapping S_T defined by $x_T(t) = x(t)$ for $0 \leq t \leq T$, $x_T(t) = 0$ for $t > T$ satisfies the above requirements.

b) Let Ω and $\mathfrak{E}$ have the same meaning as in a); if λ is real, we may put

$$F = \{x: x \in \tilde{F}, x \text{ continuous on } [0, \infty)\},$$

$$F^* = \{x: x \in F \text{ and } e^{\lambda t} |x(t)| \text{ is bounded on } [0, \infty)\},$$

$$\|x\| = \sup_{[0, \infty)} e^{\lambda t} |x(t)| \text{ for } x \in F^*,$$

$$x_T(t) = x(t) \text{ for } 0 \leq t \leq T$$

$$= x(T) e^{\lambda(T-t)} \text{ for } t > T.$$

c) Let $\Omega = \{1, 2, 3, \dots\}$ and $\mathfrak{E} = R^1$; then $\tilde{F}$ is the set of all infinite sequences of numbers. If $1 \leq p < \infty$, we may put $F = \tilde{F}$, $F = l_p$, i.e.

$$l_p = \{x: x \in \tilde{F}, \sum_{i=1}^\infty |x(i)|^p < \infty\} \text{ with usual norm and}$$

$$x_T(t) = x(t) \text{ for } t = 1, 2, \dots, T, \\ = 0 \text{ for } t > T.$$

Next, let α be a fixed nonnegative function defined on Ω such that $\alpha(t) \rightarrow 0$ as $t \rightarrow \infty$.

Let

$$F^\alpha = \{x: x \in F \text{ and } \limsup_{T \rightarrow \infty} \alpha(T) \|x_T\| < \infty\}, \text{ and for } x \in F^\alpha, \text{ put} \quad (1)$$

$$\|x\| = \limsup_{T \rightarrow \infty} \alpha(T) \|x_T\|. \quad (2)$$

(Here, $\limsup_{T \rightarrow \infty} \varphi(T)$ stands for $\lim_{T \rightarrow \infty} \{\sup_{\tau \in \Omega - [T]} \varphi(\tau)\}$.)

It is clear that $F^* \subset F^\alpha \subset F$, and that F^α is a linear set with $\|\cdot\|$ being a seminorm on F^α .

Furthermore, let

$$F^{\alpha 0} = \{x: x \in F^\alpha \text{ and } \|x\| = 0\}; \quad (3)$$

Then, obviously $F^{\alpha 0}$ is a linear set and $F^* \subset F^{\alpha 0} \subset F^\alpha$.

Let us now introduce the following concepts of continuity.

Let A be an operator mapping F into itself; A will be called E -continuous at a point $x \in F$, if for every $\epsilon > 0$ a $\delta > 0$ exists such that, for any $\tilde{x} \in F$ with $\tilde{x} - x \in F^\alpha$ and $\|\tilde{x} - x\| < \delta$, we have $A\tilde{x} - Ax \in F^\alpha$ and $\|A\tilde{x} - Ax\| < \epsilon$. The operator A will be called E -continuous, if it is E -continuous at every point of F .

Moreover, A will be called E_0 -continuous at $x \in F$, if $y \in F$ and $y - x \in F^{\alpha 0}$ imply that $Ay - Ax \in F^{\alpha 0}$.

It is easy to verify that the following proposition is true:

If A is E -continuous, then it is E_0 -continuous at every point $x \in F$.

Now, if the input-output behavior of a dynamical system $\mathcal{U}$ is described by an operator $A: F \rightarrow F$ and A is E -continuous, then $\mathcal{U}$ will be called energetically stable.

The physical interpretation of these concepts is straightforward. If $x \in F^\alpha$, the value $\|x\|$ may be interpreted as an average power of the quantity x in the time-span $\inf \Omega, \infty$. Then the introduced energetic stability characterizes a certain insensitivity of the system with respect to the average power of the signal.

Observe also the following fact: if we specify the sets $\tilde{F}, F, F^*$ as in the above example (a) with $n = 1, p = 2$ and put $\alpha(t) = t^{-2}$ for $t > 1$, then $F^\alpha \supset E$, where E is the set considered in the paper [1]. In particular, we have $x \in F^{\alpha 0}$ iff $\lim_{T \rightarrow \infty} \left(\frac{1}{T} \int_0^T |x(t)|^2 dt \right)^{1/2} = 0$. Thus, if the operator A de-

scribing a dynamical system satisfies the condition $A\theta = \theta$, then the concept of energetic stability defined in [1] coincides with our energetic stability corresponding to E_0 -continuity of A at $x = \theta$.

Before turning to the results concerning the energetic stability of feedback systems, let us make the following comment: the analysis of a feedback system reduces in essence to analysis of a functional equation. To be more specific, the existence, uniqueness and certain properties of the solution imply the existence of the over-all transfer operator and the input-output stability, respectively. A more thorough discussion may be found in [2]. Consequently, we are going to focus our attention on equations of this type. In the general case, the equation in question has the form

$$x = A(u, x); \quad (4)$$

here, A is an operator mapping $F \times F$ into F , which is specified by the system considered, u is determined by the signal at the input and x is the sought quantity determining the response of the system.

If, for each $u \in F$, a uniquely determined $x \in F$ exists such that (4) is satisfied, equation (4) defines an operator Q from F into itself by $Qu = x$; in this case we will say that (4) has the resolvent operator Q . In further

considerations we will deal only with systems which are described by causal operators, i.e.:

An operator $B: F \rightarrow F$ is called causal, if $S_T B = S_T B S_T$ for every $T \in \Omega$.

In the sequel, the following auxiliary proposition will be needed:

Lemma 1. Let B be a causal operator mapping F into itself; if, for each $T \in \Omega$, there exists a unique $x^{(T)} \in F^*$ satisfying the equation

$$x^{(T)} = S_T B x^{(T)}, \quad (5)$$

then there exists a unique $x \in F$ satisfying the equation

$$x = Bx. \quad (6)$$

Moreover, we have $S_T x = x^{(T)}$ for every $T \in \Omega$. (For the proof see [2].)

Now, all preliminaries are assembled for stating the first theorem.

Theorem 1. Let A be an operator mapping $F \times F$ into F , and let the following conditions be satisfied:

1. The operator $\tilde{A}_u = A(u, \cdot)$ is causal for any $u \in F$.

2. A number $\lambda < 1$ and an integer $m \geq 1$ exist such that

$$\|S_T(\tilde{A}_u^{m-1} y_1 - \tilde{A}_u^{m-1} y_2)\| \leq \lambda \|S_T(y_1 - y_2)\| \quad (7)$$

for all $u, y_1, y_2 \in F$ and $T \in \Omega$.

3. The operator $\tilde{A}_u^m$ is E -continuous for every $v \in F$. Then the equation

(4) has a resolvent operator Q and Q is E -continuous.

Proof: Choose $u \in F$ and let $T \in \Omega$; since F^* is a Banach space, condition

(7) shows that there exists a unique $x^{(T)} \in F^*$ such that

$$x^{(T)} = S_T \tilde{A}_u^m x^{(T)}. \quad (8)$$

We are going to show that $x^{(T)}$ is simultaneously a unique solution of the equation

$$x^{(T)} = S_T \tilde{A}_u x^{(T)}. \quad (9)$$

Actually, starting from the causality of $\tilde{A}_u$ it is not hard to verify that $\tilde{A}_u^m$ is also causal and that $S_T \tilde{A}_u^m = B^m$ with $B = S_T \tilde{A}_u$. Thus, (8) yields $Bx^{(T)} = B(B^m x^{(T)}) = B^m(Bx^{(T)})$, and consequently, due to the uniqueness of $x^{(T)}$, $Bx^{(T)} = x^{(T)}$. Hence, $x^{(T)}$ satisfies (9). Conversely, if $y \in F^*$ satisfies (9), i.e. $y = By$, it follows that $y = B(By) = B^2 y = \dots = B^m y$, so that necessarily $y = x^{(T)}$.

Thus, invoking Lemma 1, there exists a unique $x \in F$ such that

$$x = \tilde{A}_u x = A(u, x) \quad (10)$$

and $S_T x = x^{(T)}$ for any $T \in \Omega$. Hence, (4) has a resolvent operator Q .

Next, choose a fixed $u \in F$ and $\varepsilon > 0$; further, let $\tilde{u} \in F$ be such that $\tilde{u} - u \in F^0$, let $Q\tilde{u} = x = \tilde{A}_u x$, $Q\tilde{u} = \tilde{x} = \tilde{A}_{\tilde{u}} \tilde{x}$ and let $T \in \Omega$. Then we have by (8) and (7),

$$\|\tilde{x}^{(T)} - x^{(T)}\| \leq \|S_T(\tilde{A}_{\tilde{u}}^m x^{(T)} - \tilde{A}_{\tilde{u}}^m x^{(T)})\| + \|S_T(\tilde{A}_{\tilde{u}}^m x^{(T)} - \tilde{A}_u^m x^{(T)})\| \leq$$

$$\lambda \|S_T(\tilde{x}^{(T)} - x^{(T)})\| + \|S_T(\tilde{A}_{\tilde{u}}^m x^{(T)} - \tilde{A}_u^m x^{(T)})\|;$$

consequently,

$$\|S_T(\tilde{x} - x)\| \leq (1 - \lambda)^{-1} \|S_T(\tilde{A}_{\tilde{u}}^m x - \tilde{A}_u^m x)\|. \quad (11)$$

Because $\tilde{A}_{\tilde{u}}^m x$ is E -continuous by assumption 3, there exists a $\delta > 0$ such that for $\|\tilde{u} - u\| < \delta$ we have $g = \tilde{A}_{\tilde{u}}^m x - \tilde{A}_u^m x \in F^0$ and $\|g\| < (1 - \lambda)\varepsilon$, i.e.

$$\limsup_{T \rightarrow \infty} \alpha(T) \|S_T g\| < (1 - \lambda) \varepsilon.$$

Thus, (11) yields for any $\tau \in \Omega$,

$$\sup_{T \in \Omega - [\tau]} \alpha(T) \|S_T(\tilde{x} - x)\| \leq (1 - \lambda)^{-1} \sup_{T \in \Omega - [\tau]} \alpha(T) \|S_T g\|,$$

and consequently,

$$\|\tilde{x} - x\| \leq (1 - \lambda)^{-1} \|g\| < \varepsilon.$$

Hence, $Q\tilde{u} - Q\tilde{u} \in F^0$ and $\|Q\tilde{u} - Q\tilde{u}\| < \varepsilon$, i.e. Q is E -continuous; the theorem is proved.

Consider now a quasi-linear case of equation (4); here we have the proposition

Theorem 2. Let A_1 and C be operators mapping F into itself, let C be linear, causal and such that $I-C$ is one-to-one from F onto F and $(I-C)^{-1}$ is causal; furthermore, let $\tilde{A}$ map $F \times F$ into F and let $\tilde{A}(u, \cdot)$ be causal for every $u \in F$. If

1. C maps F^* into itself and $(I-C)^{-1}$ is bounded on F^* ,

2. A constant $d > 0$ exists such that

$$\|S_T(\tilde{A}(u, v_1) - \tilde{A}(u, v_2))\| \leq d \|S_T(v_1 - v_2)\| \quad (12)$$

for all $u, v_1, v_2 \in F$ and $T \in \Omega$,

3. $\|(I-C)^{-1}\|d < 1$

4. A_1 and, for every $v \in F$, $\tilde{A}(\cdot, v)$ are E -continuous, then the equation (4) with $A(u, v) = A_1 u + Cv + \tilde{A}(u, v)$ has a resolvent operator Q and Q is E -continuous.

Proof: Referring to Lemma 1, consider the equation

$$x^{(T)} = S_T A(u, x^{(T)}) = \quad (13)$$

$$S_T \{A_1 u + Cx^{(T)} + \tilde{A}(u, x^{(T)})\}.$$

By the assumption on causality of C and $(I-C)^{-1}$ it follows that (13) is equivalent to (note that $S_T x^{(T)} = x^{(T)}$)

$$x^{(T)} = S_T R_u x^{(T)}, \quad (14)$$

where

$$R_u x^{(T)} = (I - C)^{-1} \{A_1 u + \tilde{A}(u, x^{(T)})\}. \quad (15)$$

Conditions 2, 3 and axiom (iv) show that (14) has a unique solution $x^{(T)}$ in F^* , and consequently, (4) has a unique solution in F . Hence, (4) has a resolvent operator Q by Lemma 1.

Next, choose a fixed $u \in F$ and $\epsilon > 0$; if $\tilde{u} \in F$ is such that $\tilde{u} - u \in F^\alpha$ and $Qu = x = A(u, x)$, $Q\tilde{u} = \tilde{x} = A(\tilde{u}, \tilde{x})$, then it follows as in the proof of Theorem 1 that, for any $T \in \Omega$,

$$\|S_T(\tilde{x} - x)\| \leq (1 - \tilde{\alpha})^{-1} \|S_T(R_{\tilde{u}} x_T - R_u x_T)\| \text{ with } \tilde{\alpha} = \|(I - C)^{-1}\|d. \quad (16)$$

However, by (15) and the assumed causality of $\tilde{A}(u, \cdot)$,

$$\|S_T(R_{\tilde{u}} x_T - R_u x_T)\| \leq \quad (17)$$

$$\mu \|S_T(A_1 \tilde{u} - A_1 u)\| + \mu \|S_T\{\tilde{A}(\tilde{u}, x) - \tilde{A}(u, x)\}\|,$$

where $\mu = \|(I-C)^{-1}\|$.

By assumption on E -continuity of A_1 and $\tilde{A}(\cdot, v)$ there exists a $\delta > 0$ such that, for $\|\tilde{u} - u\| < \delta$, we have $A_1 \tilde{u} - A_1 u \in F^\alpha$, $\tilde{A}(\tilde{u}, x) - \tilde{A}(u, x) \in F^\alpha$ and $\|A_1 \tilde{u} - A_1 u\| < \frac{1}{2}(1 - \tilde{\alpha})\mu^{-1}\epsilon$, $\|\tilde{A}(\tilde{u}, x) - \tilde{A}(u, x)\| < \frac{1}{2}(1 - \tilde{\alpha})\mu^{-1}\epsilon$. Using (16), (17) and the same procedure as in the proof of Theorem 1 we easily conclude that $Q\tilde{u} - Qu \in F^\alpha$ and $\|Q\tilde{u} - Qu\| < \epsilon$. Hence, the proof.

In practice the operator A has frequently the form $A(u, v) = u + CNv$, where C is linear. Here, we have the following proposition, analogous to a theorem in [3]:

Theorem 3. Let C and N be causal operators mapping F into itself, and let C be linear; furthermore, let a number λ exist such that the following conditions are met:

1. $I - \lambda C$ is one-to-one from F onto F , $(I - \lambda C)^{-1}$ is causal and E -continuous.

2. $(I - \lambda C)^{-1}C$ is bounded on F^* .

3. There exists a number $\mu > 0$ such that

$$\|S_T\{Nx_1 - Nx_2 - \lambda(x_1 - x_2)\}\| \leq \mu \|S_T(x_1 - x_2)\| \quad (18)$$

for all $x_1, x_2 \in F$ and $T \in \Omega$.

4. $\|(I - \lambda C)^{-1}C\|\mu < 1$.

Then the equation $x = u + CNx$ has a resolvent operator Q and Q is E -continuous.

Proof: Referring to Lemma 1 consider the equation

$$x^{(T)} = S_T(u + CN x^{(T)}) \quad (19)$$

on F^* . By assumption 1, it follows that (19) is equivalent to

$$x^{(T)} = Rx^{(T)}, \quad (20)$$

where

$$Ry = S_T(I - \lambda C)^{-1}u + S_T(I - \lambda C)^{-1}C(N - \lambda I)y. \quad (21)$$

Due to 1 and 2 we have for any $y_1, y_2 \in F$,

$$\|Ry_1 - Ry_2\| = \|S_T\{(I - \lambda C)^{-1}C[S_T\{Ny_1 - Ny_2 - \lambda y_1 + \lambda y_2\}]\|\leq$$

$$\|(I - \lambda C)^{-1}C\|\mu\|S_T(y_1 - y_2)\|.$$

This together with 4. shows that (20) has a unique solution in F^* . Thus, by Lemma 1, the equation $x = u + CNx$ has a resolvent operator Q .

Employing the same argument as before, we obtain for any $\tilde{u}, u \in F, T \in \Omega$,

$$\|S_T(Q\tilde{u} - Qu)\| < (1 - g)^{-1} \|S_T(I - \lambda C)^{-1}(\tilde{u} - u)\|$$

with $g = \|(I - \lambda C)^{-1}C\|\mu$. Realizing the fact that $(I - \lambda C)^{-1}$ is E -continuous by 1., the rest of the proof is obvious.

Concluding the paper let us present a realistic example. Consider the feedback system portrayed in Fig. 1., where C signifies a linear time invariant system and N a pure memoryless gain. (See Fig. 1). Let these systems be governed by equations

$$z = Cw = a_w + \int_0^t k(t - \tau) w(\tau) d\tau \quad (22)$$

$$v = N\dot{z} = f(y), \quad (23)$$

where a is a real constant $n \times n$ matrix, $k(t)$ is a real $n \times n$ matrix-function defined on $[0, \infty)$ and f is an n -vector-valued function of an n -vector argument. Let the signals and responses be interpreted as elements from the set $F = \bar{L}_2$ discussed in example a). Furthermore, assume that the following conditions are met:

(i) Numbers λ and $\mu > 0$ exist such that

$$|f(\xi_1) - f(\xi_2) - \lambda(\xi_1 - \xi_2)| \leq \mu \|\xi_1 - \xi_2\| \quad (24)$$

for any $\xi_1, \xi_2 \in R^n$. (Here $\|\cdot\|$ signifies the usual norm on R^n).

(ii) The integral $K(p) = \int_0^\infty k(t) e^{-pt} dt$ converges for $\text{Re } p > -\epsilon$, $\epsilon > 0$, and $K(p)$ has rational functions of p as its elements.

(iii) $\det(I - \lambda a - \lambda K(p)) \neq 0$ in the half-plane $\text{Re } p > -\epsilon$.

We are to find constraints guaranteeing the energetic stability of the system.

The considered feedback system is governed by the equations

$$y = C(u + \psi), \quad \psi = Ny, \quad (25)$$

where u, y stands for the input signal and the output response, respectively. Thus, we have for y ,

$$y = \tilde{u} + C Ny, \quad (26)$$

where $\tilde{u} = Cu$.

Referring to Theorem 3 consider the operators $C, I - \lambda C$ and $(I - \lambda C)^{-1}C$. First, condition (iii) implies that the matrix $I - \lambda a$ is nonsingular, since $K(p) \rightarrow 0$ as $p \rightarrow \infty, p$ real. Consequently, the operator $I - \lambda C$ is one-to-one from $\bar{L}_2$ onto $\bar{L}_2$, and because $(I - \lambda C)^{-1}$ is also a Volterra-type operator, $(I - \lambda C)^{-1}$ is causal.

Using the arguments developed in [4], we can show that $I - \lambda C$ is one-to-one from $\bar{L}_2$ onto $\bar{L}_2$ and that both $I - \lambda C$ and $(I - \lambda C)^{-1}$ are bounded on $\bar{L}_2$. Thus, conditions 1. and 2. in Theorem 3 are met. (Note that the E -continuity of $(I - \lambda C)^{-1}$ is a trivial consequence of boundedness on $\bar{L}_2$). Moreover, a standard routine shows that (24) implies condition (18).

Finally, if M is an $n \times n$ matrix, let $\Lambda(M)$ signify the square-root of the largest eigenvalue of the matrix $\bar{M}M$. (The symbol $\bar{M}$ designates the complex-conjugate transpose of M). If $\eta \in L_2$, let $\hat{\eta}$ stand for the Fourier-Plancherel transform of η . Denote $Z(p) = (I - \lambda a - \lambda K(p))^{-1}(a + K(p))$ and let $z \in L_2$, $u = (I - \lambda C)^{-1}Cz$. Using the facts mentioned above and Parseval's equality, we can write

$$\|u\|^2 = \int_0^\infty \bar{u} u \, dt = \frac{1}{2\pi} \int_{-\infty}^\infty \bar{\hat{u}} \hat{u} \, d\omega = \frac{1}{2\pi} \int_{-\infty}^\infty \bar{\hat{Z}} \hat{Z}(i\omega) \hat{z}(i\omega) \hat{z} \, d\omega \leq$$

$$\frac{1}{2\pi} \sup_{\omega \in (-\infty, \infty)} \Lambda^2(\hat{Z}(i\omega)) \int_{-\infty}^\infty \bar{\hat{z}} \hat{z} \, d\omega = \sup_{\omega \in (-\infty, \infty)} \Lambda^2(\hat{Z}(i\omega)) \|z\|^2.$$

Consequently, $\|(I - \lambda C)^{-1}C\| \leq \sup_{\omega \in (-\infty, \infty)} \Lambda(\hat{Z}(i\omega))$. Thus, by condition 4 in Theorem 3, the inequality

$$\mu \sup_{\omega \in (-\infty, \infty)} \Lambda \{(I - \lambda a - \lambda K(i\omega))^{-1}(a + K(i\omega))\} < 1 \quad (27)$$

is the sought requirement for the E-continuity of the resolvent operator Q for the equation (26). Because $y = QCu$ and C is obviously E-continuous (27) guarantees the energetic stability of the considered feedback system.

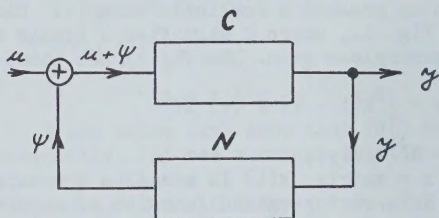


Fig. 1.

REFERENCES

- [1] Kudrewicz, J., Stability of nonlinear feedback systems (in Russian), *Avtomatika i Telemekhanika*, Vol. 25 (1964), p. 1145-1155.
- [2] Dolezal, V., On general nonlinear and quasilinear unanticipative feedback systems, to appear in *Aplikace Matematiky*.
- [3] Sandberg, I.W., Some results on the theory of physical systems governed by nonlinear functional equations, *B.S.T.J.*, Vol. 44 (1965), p. 871-898.
- [4] Sandberg, I.W., On the L_2 -boundedness of solutions of nonlinear functional equations, Vol. 43 (1964), P. 1581-1600.

STABLE INTEGRATION METHODS FOR ELECTRONIC CIRCUIT ANALYSIS WITH WIDELY SEPARATED TIME CONSTANTS

R. H. ALLEN and C. POTTLE
School of Electrical Engineering
Cornell University
Ithaca, New York

ABSTRACT

Most numerical integration rules require for stability that the integration time step be governed by the largest and possibly most uninteresting eigenvalue. Several classes of integration rules are examined to discover rules that have no numerical instabilities regardless of the step size. A class of Rosenbrock rules which possess this stability property is defined.

I. INTRODUCTION

Computer aided electronic circuit design requires fast, accurate numerical integration methods for solving the nonlinear differential representation $\dot{\underline{x}} = \underline{f}(\underline{x}, \dot{\underline{x}})$. Most of the integration methods now in use require that the integration time step size, T , be restricted so that numerical instabilities are not caused by large parasitic eigenvalues.

Several classes of integration methods are here investigated for numerical stability when the system being analyzed is stable.

II. A-STABLE INTEGRATION RULES

A generalization of Dahlquist's definition of A-stability [1] is the following: An integration rule is A-stable if all solutions of the difference equation tend to zero as $n \rightarrow \infty$, when the method is applied with fixed positive T to any differential equation of the form, $\dot{\underline{x}} = \underline{s}\underline{x}$ where s is a complex constant with negative real part.

End Point Integration Rules

As was shown by Dahlquist, no explicit integration rules are A-stable. Explicit rules have the form shown in equations (1) and (2) and include the Adams-Bashforth formulas. They are characterized by the non-appearance of derivatives of x_{n+1} on the right hand side of the formula.

$$x_{n+1} = x_n + g(x_n, x_{n-1}, \dots, \dot{x}_n, \dot{x}_{n-1}, \dots, \ddot{x}_n, \ddot{x}_{n-1}, \dots) \quad (1)$$

$$\text{e.g., } x_{n+1} = x_n + T \dot{x}_n \quad \text{Forward Euler} \quad (2)$$

Furthermore, no implicit rules relying on information previous to the n th step are A-stable. Implicit rules have the form shown in equation (3) and (4) and include the Adams-Moulton formulas. They are characterized by the appearance of derivatives of x_{n+1} on the right hand side of the formula.

$$x_{n+1} = x_n + g(x_n, x_{n-1}, \dots, \dot{x}_{n+1}, \dot{x}_n, \dot{x}_{n-1}, \ddot{x}_{n+1}, \dots) \quad (3)$$

$$\text{e.g., } x_{n+1} = x_n + \frac{T}{12} (5\dot{x}_{n+1} + 8\dot{x}_n - \dot{x}_{n-1}) \quad \text{Third-order Adams-Moulton} \quad (4)$$

The only end point methods (methods requiring information only at end points of the time step) which can possibly be A-stable are those that are implicit and one-step (requiring information only at the end points of the current

interval). A few such rules are given by equations (5) - (8). Higher order rules require higher derivatives.

$$x_{n+1} = x_n + T \dot{x}_{n+1} \quad \text{Backward Euler} \quad (5)$$

$$x_{n+1} = x_n + \frac{T}{2} (\dot{x}_{n+1} + \dot{x}_n) \quad \text{Trapezoidal} \quad (6)$$

$$x_{n+1} = x_n + T(a_1 \dot{x}_{n+1} + a_2 \dot{x}_n) + T^2(a_3 \ddot{x}_{n+1} + a_4 \ddot{x}_n) \quad \text{with one } a_i=0 \quad (7)$$

$$x_{n+1} = x_n + \frac{T}{2}(\dot{x}_{n+1} + \dot{x}_n) + \frac{T^2}{12}(-\ddot{x}_{n+1} + \ddot{x}_n) \quad (8)$$

A useful method for investigating the numerical stability of an integration rule is as follows [2]: Remove all derivative terms in the difference equation form of an integration rule using the test system

$$\dot{x} = s x, \quad (9)$$

then apply the z-transform to obtain a mapping rule $z(s)$. The region of numerical stability of the integration rule is then obtained by mapping the unit circle $|z(s)| = 1$ to the s-plane. If the function $z(s)$ maps $|z| > 1$ into the right half s-plane, with at most a simple mapping to the imaginary axis, the integration rule is A-stable.

Rosenbrock Integration Rules

Rosenbrock has presented an interesting general rule for the numerical integration of ordinary differential equations [3]. The form of this rule appears below in equations (10).

$$\underline{k}_n = T(I - T a_{1-x} f(\underline{x}_n))^{-1} \underline{f}(\underline{x}_n) \quad (10)$$

$$\underline{l}_n = T(I - T a_{2-x} \underline{f}(\underline{x}_n + c_{1-k} \underline{k}_n))^{-1} \underline{f}(\underline{x}_n + b_{1-k} \underline{k}_n)$$

$$\underline{m}_n = T(I - T a_{3-x} \underline{f}(\underline{x}_n + c_{2-k} \underline{k}_n + e_{1-l} \underline{l}_n))^{-1} \underline{f}(\underline{x}_n + b_{2-k} \underline{k}_n + d_{1-l} \underline{l}_n)$$

⋮

$$\underline{x}_{n+1} = \underline{x}_n + R_{1-k} \underline{k}_n + R_{2-l} \underline{l}_n + R_{3-m} \underline{m}_n + \dots$$

As is easily seen, this rule is neither implicit (the last step is explicit), nor one-step (intermediate points are found). One of its advantages is that intermediate points are found implicitly using information which is already known, so iteration to find $\underline{f}$, if necessary, does not require evaluation of a Jacobian matrix or an inverse at every iteration step. Higher order (>2) implicit one-step end point methods require the inversion of a Jacobian at a point which has not yet been located, demanding a difficult iteration procedure including these time-consuming operations. Another advantage is that A-stable Rosenbrock rules exist. The Rosenbrock rule of order 2 is the Trapezoidal Rule, equation (6). Calahan has presented a third order Rosenbrock rule which is the only one to possess the A-stable property [4]. Higher order A-stable rules are presented below.

III. HIGH ORDER ROSENBRICK METHODS

A particular rule is defined by giving unique values to the $a_i, b_i, \dots, g_i$ as given in equations (10). An underdetermined set of equations which these quantities must satisfy, and from which values may be assigned, is obtained by equating multinomial terms in T and f in the Taylor expansion

$$x_{n+1} \approx x_n + T\dot{x}_n + \frac{T^2}{2!}\ddot{x}_n + \dots, \dot{x}_n = f(x_n) \quad (11)$$

with the last of equations (10) in the scalar case

$$x_{n+1} = x_n + R_1 k + R_2 l + R_3 m + R_4 n, \quad (12)$$

the order of the desired method determining the highest power of T kept. The quantities k , l , m , and n are removed using the remaining equations. Inverse terms are removed by expansions in power series, e.g., (scalar case)

$$(1 - Ta_1)^{-1} \approx 1 + Ta_1 A + T^2 a_1^2 A^2 + \dots \quad (13)$$

Functions and Jacobians evaluated at intermediate points are expanded in Taylor series, e.g., (scalar case)

$$f(x_n + b_1 k) \approx f(x_n) + f'(x_n)b_1 k + \dots + \frac{f^{(5)}(x_n)}{5!}b_1^5 k^5. \quad (14)$$

Although the problem of the power series and Taylor expansions, together with coefficient matching seems ideally suited to computer solution, the available symbol manipulation programs are not equipped to handle a problem of this size. Before collecting terms the sixth order expansions have over 20,000 terms, and intermediate storage at least as great is required during the calculation. Each of these terms consists of up to five variables, six occurrences of f and its derivatives, a constant, and a power of T . The resulting polynomials contain over a quarter of a million pieces of information. Selective partitioning of the problem and naming of intermediate variables reduces the amount of storage required and simplifies the collection of terms. A special purpose FORTRAN program using 2-byte variables then carried out the symbol manipulation in a reasonable amount of time and space. The results of this program, giving the equation to be satisfied and the truncation error terms obtained, are tabulated in Appendix 1.

Selecting Rosenbrock Coefficients

There is no first order Rosenbrock rule since $a_1 = 0$ implies the first order integration method is explicit. A p -th order Rosenbrock method requires $p-1$ intermediate points $k, l, \dots$. A p -th order rule has $2(p-1)$ more coefficients than a $(p-1)$ -st rule. Therefore there are $p(p-1)$ coefficients for a p -th order rule. There are 2^{p-1} Rosenbrock equations that must be satisfied by the Rosenbrock coefficients. Table 1 summarizes the number of coefficients and equations for each order rule.

Table 1. Number of Rosenbrock coefficients and equations.

Order of integration rule	1	2	3	4	5	6
Number of intermediate points	-	1	2	3	4	-
Number of Rosenbrock coefficients	-	2	6	12	20	30
Number of Rosenbrock equations	-	2	4	8	16	32

The rules for orders 3, 4, and 5 have 2, 4, and 4 arbitrary coefficients respectively since there are more coefficients than equations to be satisfied. These arbitrary coefficients can be chosen to:

- 1) control numerical stability,
- 2) minimize computational effort, and
- 3) minimize the truncation error.

There are no 6-th and higher order rules of the Rosenbrock form because there are more equations to be satisfied than variables.

If $\dot{x} = f(x)$ is taken to be $\dot{x} = sx$, the z-transform of the 4-th order Rosenbrock rule,

$$x_{n+1} = x_n + R_1 k_n + R_2 l_n + R_3 m_n \quad (15)$$

$$\text{becomes } z = 1 + R_1 K + R_2 L + R_3 M \quad (16)$$

where $K = sT/(1 - a_1 sT)$

$$L = sT(1 + b_1 K)/(1 - a_2 sT)$$

$$M = sT(1 + b_2 K + d_1 L)/(1 - a_3 sT)$$

Using the Rosenbrock equations z becomes

$$z = \frac{1 + asT + bs^2T^2 + ds^3T^3}{1 + dsT + es^2T^2 + fs^3T^3} \quad (17)$$

$$\text{where } d = -(a_1 + a_2 + a_3)$$

$$a = 1 + d$$

$$e = a_1 a_2 + a_1 a_3 + a_2 a_3$$

$$b = \frac{1}{2} + d + e$$

$$f = -a_1 a_2 a_3$$

$$c = \frac{1}{6} + \frac{1}{2}d + e + f$$

Notice that z depends only on a_1 , a_2 , and a_3 and that if an a_1 is chosen as 0 to simplify the integration rule then $f = 0$ and the resulting integration rule will not be A-stable.

Appendix 2 summarizes some of the special case solutions of the Rosenbrock equations. Appendix 3 lists the coefficients and truncation error terms of several Rosenbrock integration rules. Figure 1 presents the stability regions for these solutions. The 4-th order solutions include 9, 20, 21, 24 and 31 which are A-stable. Solutions 1 and 2 are not A-stable but require only 1 Jacobian and 1 inverse for each integration step. Solution 10 has a large stability region. Only three 5-th order rules have been found. Solutions 5TH1 and 5TH2 are not A-stable but require 2 Jacobians and 2 inverses for each integration step. Solutions 26 and 5-JW map the unit circle $|z| = 1$ to the imaginary axis as do the one step implicit end point integration rules. Calahan's 3-rd order stable Rosenbrock rule is plotted for comparison. There is no real 3rd order Rosenbrock rule which maps the unit circle into the imaginary axis. Notice that very small images of the unit circle appear in some cases in addition to the principal one, and in particular that solutions 26 and 5-JW are unfortunately not theoretically A-stable since a small region of instability exists near $\sigma T = -1$.

IV. CONTROL OF TRUNCATION ERROR

There exists a class of one-step implicit integration methods including both end point and Rosenbrock types which are completely stable for any step size for linear systems with eigenvalues anywhere in the left half plane. Since these are one-step methods, they are self starting. They can therefore be easily changed between integration steps. Truncation error has conventionally been controlled by varying the integration time step size. Changing the integration rule during a numerical solution in this case can also be effectively used to control truncation error. When only the time step size can be changed, a single high order integration rule will probably be necessary. The high order methods require much computational effort but produce high accuracy, while a low order method will be considerably faster

Figure 1a. Unstable cases

Case 10 ———
 Case 2 ———
 Case 1 - - - - -
 Cases 3,4,5TH
 5TH

$$\lambda T = \sigma T + j\omega T$$

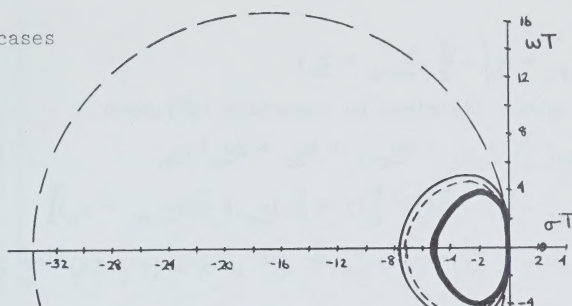


Figure 1b. Stable cases

Cases 20-25 ———
 Cases 31,9 ———
 Case Calahan ———
 Case 26,5-JW ———

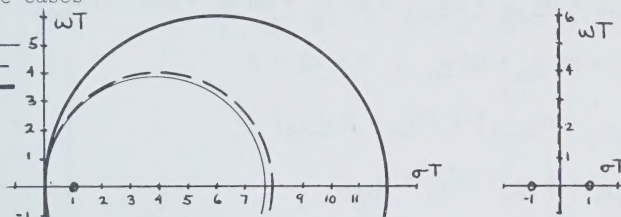


Figure 1. Stability Region in the λT Plane for Some Rosenbrock Rules

when the numerical solution is smooth enough.

Changing the order of the integration rule instead of the step size is particularly advantageous in the analysis of a nonlinear system with few actual nonlinearities. A useful approach in this case is to separate the system into its linear and nonlinear parts as shown in figure 2. [5].

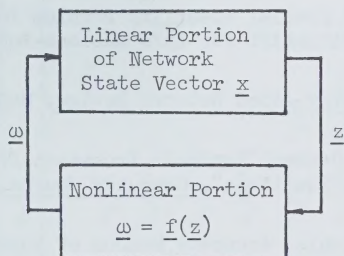


Figure 2. Nonlinear Systems Separated into Linear and Nonlinear Parts

The state equations for this system may be written

$$\dot{\underline{x}} = \underline{A} \underline{x} + \underline{E} \underline{w}$$

$$\underline{z} = \underline{G} \underline{x} + \underline{J} \underline{w}$$

$$\underline{w} = \underline{f}(\underline{z})$$

(18)

The numerical solution to these differential equations will now involve some approximation to $\exp(\lambda T)$, which need be found only once provided the step size is not changed. The saving in effort is best illustrated by an example. Applying the Trapezoidal Rule (Rosenbrock rule of order 2)

$$\underline{x}_{n+1} = \underline{x}_n + \frac{T}{2} (\dot{\underline{x}}_{n+1} + \dot{\underline{x}}_n) \quad (19)$$

to the system described by equations (18), then

$$\underline{x}_{n+1} = \frac{T}{2} (A_{\underline{x}} \underline{x}_{n+1} + E_{\underline{w}} \underline{w}_{n+1} + A_{\underline{x}} \underline{x}_n + E_{\underline{w}} \underline{w}_n) + \underline{x}_n \quad (20)$$

$$\text{or } \underline{x}_{n+1} = (I - \frac{T}{2} A)^{-1} \left[(I + \frac{T}{2} A) \underline{x}_n + \frac{T}{2} E (\underline{w}_{n+1} + \underline{w}_n) \right] \quad (21)$$

$$\text{Let } P = (I - \frac{T}{2} A)^{-1} (I + \frac{T}{2} A) \approx e^{AT}; \quad M = (I - \frac{T}{2} A)^{-1} \frac{T}{2} E \quad (22)$$

$$\text{Then } \underline{z}_{n+1} = G_{\underline{x}} \underline{x}_{n+1} + J_{\underline{w}} \underline{w}_{n+1} = G P \underline{x}_n + G M \underline{w}_n + (G M + J) \underline{w}_{n+1} \quad (23)$$

$$\text{Let } \underline{r}_n = G P \underline{x}_n + G M \underline{w}_n; \quad N = G M + J$$

$$\text{Then } \underline{w}_{n+1} = f(\underline{z}_{n+1}) = f(\underline{r}_n + N \underline{w}_{n+1}) \quad (24)$$

$$\text{and } \underline{x}_{n+1} = P \underline{x}_n + M(\underline{w}_n + \underline{w}_{n+1}) \quad (25)$$

Note that $J = 0$ implies that the general system $\dot{\underline{x}} = f(\underline{x}, \dot{\underline{x}})$ can be written $\dot{\underline{x}} = f(\underline{x})$, and no iteration is required to find $\underline{w}$ at each step. P is the (stable) Pade' approximation to $\exp(AT)$ [2]. If $\underline{w}$ is of size p and $\underline{x}$ is of size m , then conventionally there are m^3 operations per integration step from the required large Jacobian, large matrix inversion, and large iteration to solve for $\underline{x}$. By this separated approach if $p \ll m$ then the dominating number of operations per integration step is $p^3 + m^2 \approx m^2$.

REFERENCES

1. Germund G. Dahlquist, "A Special Stability Problem for Linear Multistep Methods," BIT, Nordisk Tidskrift for Informations-Behandling, Vol.3, 1963, pp. 27-43.
2. Donald A. Calahan, Computer-Aided Network Design, McGraw-Hill Book Company, 1968, pp. 62-75.
3. H. H. Rosenbrock, "Some General Implicit Processes for the Numerical Solution of Differential Equations", Computer Journal, Jan. 1963, Vol.5, No.4, pp. 329-330.
4. Donald A. Calahan, "A Stable, Accurate Method of Numerical Integration for Nonlinear Systems", IEEE Proceedings, April 1968, p. 744.
5. R. H. Allen and C. Pottle, "High Speed Digital Computer Analysis of Non-linear and Time-Varying Electronic Circuits", Proceeding of the Fifth Annual Allerton Conference on Circuit and System Theory, University of Illinois, October 1967, pp. 534-543.

APPENDIX I. Rosenbrock Equations and Truncation Error Terms

To determine the coefficients for a p-th order Rosenbrock integration method, the set of Rosenbrock equations $f_i = R_1 k_i + R_2 \ell_i + R_3 m_i + R_4 n_i$ must be satisfied. The truncation error constants are given by $e_j = R_1 k_j + R_2 \ell_j + R_3 m_j + R_4 n_j$.
 Table A1. Index range and conditions for Rosenbrock equations and error terms.
 Order Equations f_i Error constants e_j Conditions Order Equations f_i Error constants e_j Conditions

2	$i = 0, 1$	$j = 2, 3$	$R_2 + R_3 + R_4 = 0$	4	$i = 0, \dots, 7$	$j = 8, \dots, 15$	$R_4 = 0$
3	$i = 0, \dots, 3$	$j = 4, \dots, 7$	$R_3 = R_4 = 0$	5	$i = 0, \dots, 15$	$j = 16, \dots, 31$	

Table A2. Intermediate terms for Rosenbrock equations and Error Terms

i	f_i	k_i	ℓ_i	m_i	n_i	Intermediate variables
0	1	1	1	1	1	$b = b_2 + d_1$
1	$\frac{1}{2}$	a_1	a_2	$a_3 + b$	a_4	$c = c_2 + e_1$
2	$\frac{1}{6}$	a_1^2	$a_2 \ell_1 + a_3 b_1$	$a_3 m_1 + u_1$	$a_4 n_1 + t_1$	$d = b_2 + d_2 + h_1$
3	$\frac{1}{6}$	$a_2 c_1 + \frac{1}{2} b_1^2$	$a_2 c_1 + \frac{1}{2} b_1^2$	$a_3 c + \frac{1}{2} b_1^2$	$a_4 e + \frac{1}{2} d^2$	$e = c_2 + e_2 + g_1$
4	$\frac{1}{24}$	a_1^3	$a_2 \ell_2 + a_3 b_1^2$	$a_3 m_2 + u_1$	$a_4 n_2 + t_1$	$u_1 = a_1 b_2 + \ell_1 d_1$
5	$\frac{1}{24}$	$a_2 \ell_3$	$a_2 \ell_3$	$a_3 m_3 + d_1 \ell_3$	$a_4 n_3 + t_6$	$u_2 = a_1 c_2 + \ell_1 e_1$
6	$\frac{1}{24}$	$a_2 c_1 (\ell_1 + a_3) + a_3 b_1^2$	$a_2 c_1 (\ell_1 + a_3) + a_3 b_1^2$	$a_3 (cm_1 + u_2) + bu_1$	$a_4 en_1 + a_4 t_2 + dt_1$	$u_3 = a_1 c_2 + \ell_2 e_1$
7	$\frac{1}{24}$	$\frac{1}{2} a_2 c_1^2 + \frac{1}{2} b_1^3$	$\frac{1}{2} a_2 c_1^2 + \frac{1}{2} b_1^3$	$\frac{1}{2} a_3 c_2 + \frac{1}{2} b_1^3$	$\frac{1}{2} a_4 e + \frac{1}{2} d^3$	$u_4 = a_1 b_2 + \ell_2 d_1$
8	$\frac{1}{120}$	a_1^4	$a_2 \ell_4 + a_3 b_1^3$	$a_3 m_4 + u_7$	$a_4 n_4 + t_7$	$u_7 = a_1 b_2 + \ell_4 d_1$
9	$\frac{1}{120}$	$a_1 \ell_6 + a_2 c_1 \ell_2 + a_3 b_1^2$	$a_1 \ell_6 + a_2 c_1 \ell_2 + a_3 b_1^2$	$a_3 cm_2 \ell_1 + a_3 (u_2 m_1 + u_3) + bu_4$	$\frac{1}{2} t_2 + a_4 (t_2 n_1 + t_2 e n_2) + dt_4$	$u_8 = a_1 c_2 + \ell_4 e_1$
10	$\frac{1}{120}$	$a_2 \ell_5$	$a_2 m_5 + d_1 \ell_5$	$a_4 n_5 + \ell_5 d_2 + m_5 h_1$	$a_4 n_5 + \ell_5 d_2 + m_5 h_1$	$t_1 = a_1 b_2 + \ell_1 d_2 + m_1 h_1$
11	$\frac{1}{120}$	$a_2 \ell_6$	$a_2 m_6 + d_1 \ell_6$	$a_4 n_6 + \ell_6 d_2 + m_6 h_1$	$a_4 n_6 + \ell_6 d_2 + m_6 h_1$	$t_2 = a_1 c_2 + \ell_1 e_2 + m_1 g_1$
12	$\frac{1}{120}$	$a_2 \ell_7$	$a_2 m_7 + d_1 \ell_7$	$a_4 n_7 + \ell_7 d_2 + m_7 h_1$	$a_4 n_7 + \ell_7 d_2 + m_7 h_1$	$t_3 = a_1 c_2 + \ell_1 e_2 + m_1 g_1$
13	$\frac{1}{120}$	$c_1 \ell_5$	$a_3 cm_5 + a_3 \ell_1 e_2 + b \ell_3 d_1$	$a_4 en_5 + a_4 t_5 + dt_6$	$a_4 en_5 + a_4 t_5 + dt_6$	$t_4 = a_1 b_2 + \ell_1 d_2 + m_1 h_1$

i	f _i	k _i	l _i	m _i	n _i	Intermediate Variables
14	$\frac{1}{120}$		$a_2 c_1^2 (l_1^2 + a_1) + \frac{1}{2} b_1^3 a_1$	$\frac{1}{2} a_2 c_1^2 m_1 + a_2 c_1 u_2 + \frac{1}{2} b_1^2 u_1$	$\frac{1}{2} a_1 e_1 n_1 + a_1 e t_2 + \frac{1}{2} d_1^2 t_1$	$t_5 = l_3 e_2 + m_3 g_1$
15	$\frac{1}{120}$		$\frac{1}{6} a_2 c_1^3 + \frac{1}{24} b_1^4$	$\frac{1}{6} a_2 c_1^3 + \frac{1}{24} b_1^4$	$\frac{1}{6} a_1 e_1^3 + \frac{1}{24} d_1^4$	$t_6 = l_3 d_2 + m_3 h_1$
16	a_1^5		$a_2 l_8 + a_1 b_1^4$	$a_2 m_8 + a_1 b_1^2 + l_8 d_1^4$	$a_1 n_8 + a_1 b_1 + l_8 d_2 + m_8 h_1$	$t_7 = a_1^3 b_1 + l_4 d_2 + m_4 h_1$
17			$a_2 l_{10}$	$a_2 m_{10} + l_{10} d_1$	$a_1 n_{10} + l_{10} d_2 + m_{10} h_1$	$t_8 = a_1^3 c_1 + l_4 e_2 + m_4 g_1$
18			$a_2 l_{11}$	$a_2 m_{11} + l_{11} d_1$	$a_1 n_{11} + l_{11} d_2 + m_{11} h_1$	
19			$a_2 l_9$	$a_2 m_9 + l_9 d_1$	$a_1 n_9 + l_9 d_2 + m_9 h_1$	
20			$c_1 l_{12}$	$a_2 c m_7 + a_2 l_7 e_1 + b l_7 d_1$	$a_1 e n_7 + a_1 l_7 e_2 + a_1 m_7 g_1 + d l_7 d_2 + d m_7 h_1$	
21			$a_2 l_{12}$	$a_2 m_{12} + l_{12} d_1$	$a_1 n_{12} + l_{12} d_2 + m_{12} h_1$	
22			$a_2 l_{14}$	$a_2 m_{14} + l_{14} d_1$	$a_1 n_{14} + l_{14} d_2 + m_{14} h_1$	
23			$\frac{1}{2} c_1 l_{13}$	$\frac{1}{2} a_2 c_1^2 m_2 + a_2 c_1 l_3 e_1 + \frac{1}{2} b_1^2 l_3 d_1$	$\frac{1}{2} a_1 e_1 n_3 + a_1 e t_5 + \frac{1}{2} d_1^2 t_6$	
24			$c_1 l_{11}$	$a_3 (c m_6 + m_1 l_3 e_1 + l_6 e_1) + b l_6 d_1 + \frac{1}{2} b_1 u_1 l_3 d_1$	$a_4 (e n_6 + t_5 n_1 + l_6 e_2 + m_5 g_1) + d l_6 d_2 + m_6 h_1 + d \frac{1}{2} t_6 t_6$	
25			$a_2 l_{13}$	$a_2 m_{13} + l_{13} d_1$	$a_1 n_{13} + l_{13} d_2 + m_{13} h_1$	
26			$a_2 c_1 (l_4 + a_1 l_2) + a_1^2 (l_6^2 + a_1 b_1^2)$	$a_3 (c m_4 + m_2 u_2 + u_2 m_1 + u_8) + u_1 u_4 + b u_7$	$n_2 a_4 (a_1 e + t_2) + a_1 t_3 n_1 + t_4 (a_1 e + t_1) + t_8 a_1 + d t_7$	
27			$a_2 c_1^2 (l_2^2 + a_1 l_1 + \frac{3}{2} a_1^2) + a_1 b_1^2$	$a_3 (c (\frac{1}{2} c m_2 + m_1 u_2 + u_3) + \frac{1}{2} b_1 u_2^2) + \frac{1}{2} b (u_1^2 + b u_4^2)$	$a_4 (e n_1 t_2 + \frac{1}{2} t_2^2 + e t_3 + \frac{1}{2} e n_2) + \frac{1}{2} c_1 d_1 + \frac{1}{2} d_1^2 t_4$	
28			$l_{25} + a_1 l_{13}$	$a_3 (c m_5 + u_2 m_2 + l_5 e_1) + \frac{1}{2} b_1 u_1 l_2 d_1 + b d_1 l_5$	$a_4 (e n_5 + t_2 n_3 + l_5 e_2 + m_5 g_1) + \frac{1}{2} t_6 t_6 + d l_5 d_2 + m_5 h_1 + d$	
29			$a_2 l_{15}$	$a_2 m_{15} + d_1 l_{15}$	$a_1 n_{15} + l_{15} d_2 + m_{15} h_1$	
30			$\frac{1}{6} a_2 c_1^3 + \frac{1}{24} b_1^4$	$\frac{1}{6} a_2 c_1^3 m_1 + \frac{1}{24} b_1^4 u_1 + \frac{1}{2} a_2 c_1^2 u_2$	$\frac{1}{6} a_1 e_1^3 n_1 + \frac{1}{24} d_1^4 t_1 + \frac{1}{2} a_1 e_1^2 t_2$	
31			$\frac{1}{24} a_2 c_1 + \frac{1}{240} b_1^5$	$\frac{1}{24} a_2 c_1 + \frac{1}{240} b_1^5$	$\frac{1}{24} a_1 e_1^4 + \frac{1}{240} d_1^5$	

APPENDIX 2

Table A3. Some Interesting Cases for the Solution of the Rosenbrock 4-th Order Equations.

Case Number of Jacobians	Number of Inversions	Fixed Coefficients	Number		Are there stable real sets?
			Real Sets	Number Found	
1	1	$a_2=a_3=.5, a_1=0, c_1=c_2, e_1=0$	4	4	no
2	1	$a_2=a_3=1/6, a_1=0, c_1=c_2, e_1=0$	4	4	no
3	1	$a_1=0, a_2=0, c_2=0, e_1=0$	0	0	no
4	1	$a_1=0, a_2=0, c_1=0$	0	0	no
5	1	$a_1=a_2=a_3, c_1=c_2=0, e_1=0$	0	0	no
6	1	$a_2=0, c_1=0, c_2=0, e_1=0$	?	0	no
7	1	$a_2=0, c_1=0, c_2=0, e_1=0$	?	0	no
8	1	$a_1=0, e_1=0, c_1=0$	?	0	no
9	1	$a_1=a_2, c_1=0, c_2=0, e_1=0$	?	1	yes
10	1	$a_1=a_3, c_1=0, c_2=0, e_1=0$	?	1	no
11	1	$a_2=a_3, c_1=0, c_2=0, e_1=0$	0	0	no
12	1	$c_1=0, c_2=0, e_1=0, b_1=0$	0	0	no
13	1	$c_1=0, c_2=0, e_1=0$	?	0	?
14	2	$a_1=0$	?	0	no
15	2	$a_2=0, c_1=0$	?	0	no
16	2	$a_2=0, c_2=0, e_1=0$	?	0	no
17	2	$a_1=a_2, c_1=0$	?	0	?
18	2	$a_2=a_3, c_1=c_2, e_1=0$	?	0	?
19	2	$a_1=a_3, c_2=0, e_1=0$	?	0	?
20	2	$a_1=a_2=a_3, c_1=0, e_1=0$	6	2	yes
21	2	$a_1=a_2=a_3, c_1=0, c_2=0$	3	1	yes
22	2	$a_1=a_2=a_3, c_1=0, c=0$	?	0	yes
23	2	$a_1=a_2=a_3, c_1=0, b=0$	?	0	yes
24	2	$a_1=a_2=a_3, e_1=0, c_1=c_2$	6	2	yes
25	2	$a_1=a_2=a_3, e_1=0, c_2=0$	?	0	yes
26	2	$a_1=a_2, e_1=0, c_1=0, .5=a_1+a_2+a_3$	6	6	yes
27	2	$c_1=0$	∞	0	yes
28	2	$c_2=0, e_1=0$	∞	0	yes
29	2	$c_1=c_2, e_1=0$	∞	0	yes
30	3		∞	0	yes

APPENDIX 3. ROSEN BROCK COEFFICIENTS AND TRUNCATION ERROR TERMS

	CASE 1A	CASE 2A	CASE 9	CASE 10	CASE 20A
R1	0.51919682	0.58289896	-1.31690856	4.48858461	-0.99974352
R2	0.10517721	-2.22345440	-0.72308665	14.24359481	1.04980048
R3	0.37562597	2.64055544	3.03599525	-17.72217942	0.54994304
A1	0.00000000	0.00000000	1.00948231	0.29494239	1.06857900
A2	0.50000000	0.16666667	1.00948231	0.31856704	1.06857900
A3	0.50000000	0.16666667	1.21823218	0.29494239	1.06857900
B1	1.1855168	0.50661594	-1.14208006	0.38512482	-1.13715813
B2	0.49255558	0.53727153	-0.51694892	0.29209618	0.88028175
C1	0.28684941	-0.08456636	0.00000000	0.00000000	0.00000000
C2	0.28684941	-0.08456636	0.00000000	0.00000000	-0.70716882
D1	-0.13201135	0.04729483	-0.13104677	0.02467280	-0.22212634
E1	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
T1	-1.64E-15	9.26E-03	1.76E-01	5.85E-03	1.73D-01
T2	1.81E-02	7.08E-03	-3.49E-01	-8.87E-03	-5.88E-01
T3	9.65E-13	9.26E-03	-1.12E-01	9.90E-03	-1.01E-01
T4	8.88E-03	5.77E-03	-2.75E-01	7.87D-03	-2.47D-01
T5	6.22D-03	9.63E-03	1.12E-01	1.13E-02	9.62E-02
T6	8.88D-03	5.77D-03	1.68D-01	-1.03E-02	2.97D-01
T7	4.87E-03	1.45E-02	2.22E-01	2.75D-02	3.53D-01
T8	9.72E-03	7.02E-03	-2.89E-02	5.62E-03	2.07D-02

APPENDIX 3. (CONTINUED) ROSENBROCK COEFFICIENTS AND ERROR TERMS

CASE 21		CASE 24A		CASE 26A		CASE 26B		CASE 31A	
R1	-3.48188133	0.07587238	0.65716260	0.38057781	0.51272115				
R2	-0.16519162	3.13461475	0.08626018	0.08207549	0.44643467				
R3	4.64707295	-2.21048712	0.25657723	0.53734670	0.04084419				
A1	1.06857900	1.06857900	0.67560360	0.67560360	1.00000000				
A2	1.06857900	1.06857900	0.67560360	0.67560360	1.00000000				
A3	1.06857900	1.06857900	-0.85120719	-0.85120719	1.25000000				
B1	-1.13715813	-0.97084944	-1.14490772	-1.14490772	-1.14285714				
B2	-0.11736864	-1.23131836	0.64921093	1.08695584	3.51579208				
C1	0.00000000	0.07554926	0.00000000	0.00000000	0.00000000				
C2	0.00000000	0.07554926	0.38054161	0.86359935	-6.90963038				
D1	-0.04540651	0.11180948	0.57810464	0.28792903	-3.51579208				
E1	0.04267416	0.00000000	0.00000000	0.00000000	4.46360290				
T1	1.73D-01	1.73D-01	-5.78D-02	-5.78D-02	1.77D-01				
T2	-3.81D-01	-4.31D-01	-7.06D-02	-7.60D-02	-6.20D-01				
T3	-1.01D-01	-1.01D-01	8.85D-02	8.85D-02	-1.15D-01				
T4	-2.47D-01	-2.28D-01	2.13D-01	2.13D-01	-2.81D-01				
T5	9.62D-02	8.15D-02	-1.06D-01	-1.06D-01	1.16D-01				
T6	1.72D-01	1.66D-01	8.36D-02	5.64D-02	5.31D-01				
T7	1.26D-01	2.82D-01	-3.87D-02	-5.71D-02	8.00D-01				
T8	-1.13D-02	-2.86D-02	2.84D-02	3.68D-02	-9.28D-02				

CASE 5TH1		CASE 5TH2		CASE 5-JW		CASE TRAP	
R1	0.56856364	0.63938293	0.86190000	R1	1.00000000		
R2	0.39491620	0.31923552	0.14993870	A1	0.50000000		
R3	0.03662261	0.04166054	-0.00614007	T1	2.50D-00		
R4	-0.00010244	-0.00027900	-0.00569871	T2	0.00D-00		
A1	0.00000000	0.00000000	0.26961054				
A2	0.25992664	0.26468015	0.26961054				

CASE 5TH1		CASE 5TH2		CASE 5-JW		CASE CAL	
A3	0.00000000	0.00000000	-0.59666439	R1	0.75000000		
A4	1.50923751	1.10148990	0.55744331	R2	0.25000000		
H1	0.96533210	1.07196491	1.76927111	A1	0.78867513		
B2	-0.48024684	0.78256678	1.77190000	A2	0.78867513		
B3	1.48265064	1.10128367	1.05851950	B1	-1.15470054		
C1	-0.19625203	-0.95918999	0.00000000	C1	0.00000000		
C2	0.00000000	0.00000000	-2.27373267	T1	-4.81D-02		
C3	1.50693233	1.10654750	1.83386600	T2	1.31D-01		
D1	0.93694745	1.00617685	1.83688274	T3	2.63D-01		
D2	1.37577800	1.08791341	1.58311589	T4	-6.41D-02		
E1	0.00000000	0.00000000	1.38448090				
E2	1.44851323	1.09890641	1.55518227				
G1	-2.67466829	0.54452307	-3.03180480				
H1	1.49009996	1.09861069	0.23854905				

T1	-1.60D-03	-7.63D-04	-6.19D-03
T2	5.91D-05	-1.19D-03	-6.87D-03
T3	-5.50D-03	-2.11D-02	-4.66D-02
T4	-5.10D-03	-2.72D-02	2.63D-02
T5	-1.42D-03	-1.21D-02	-4.73D-02
T6	2.24D-03	6.97D-03	-1.90D-02
T7	-2.73D-03	6.29D-03	2.68D-02
T8	1.53D-03	2.00D-02	-1.89D-01
T9	7.40D-03	-1.69D-02	1.14D-01
T0	-4.13D-03	-2.08D-02	2.22D-02
TA	1.21D-02	9.02D-05	7.94D-02
TB	8.54D-03	6.09D-02	-5.35D-01
TC	5.89D-03	-8.94D-03	6.28D-03
TD	2.60D-03	-7.71D-04	1.55D-02
TE	-2.82D-03	2.37D-02	-2.30D-01
TF	1.44D-03	1.15D-02	-1.90D-02

EFFICIENT NUMERICAL ANALYSIS OF NON-LINEAR CIRCUITS*

D. A. CALAHAN
The University of Michigan
Ann Arbor, Michigan

INTRODUCTION

The field of computer-aided circuit design is probably unique in the number of unfulfilled promises offered to the designer. This is especially true in non-linear circuit analysis, where the dimensionality and ill-conditioning problems unknowing faced by the circuit designer with no background in numerical techniques often result in early abandonment of computer-aided design.

One of the reasons for this strange evolvement is the extent of numerical ill-conditioning in the usual (i. e., switching) circuits encountered. As this is being at least partially overcome by new numerical techniques [1], [2], [3]** and as the size of analyzable circuits have grown, dimensionality problems have suddenly been recognized. This paper reports recent experiments which demonstrate use of the most recent numerical advances to obviate both the ill-conditioning and dimensionality problems in non-linear transient analysis.

EQUATION FORMULATION

We consider solution of circuit equations of the form:

$$\underline{\tilde{S}} \dot{\underline{x}} = \underline{\tilde{A}} \underline{x} + \underline{\tilde{B}} \underline{u}_s + \underline{\tilde{B}'} \underline{w}(\underline{u}_n) \quad (1)$$

or

$$\dot{\underline{x}} = \underline{f}(\underline{x}) = \underline{\tilde{S}}^{-1} \{ \underline{\tilde{A}} \underline{x} + \underline{\tilde{B}} \underline{u}_s + \underline{\tilde{B}'} \underline{w}(\underline{u}_n) \} \quad (2)$$

$$\underline{u}_n = \underline{\tilde{E}} \underline{x} + \underline{\tilde{F}} \underline{u}_s + \underline{\tilde{F}'} \underline{w}(\underline{u}_n) \quad (3)$$

where all matrices (denoted by $\sim$) are constant except $\underline{\tilde{S}}$, which contains the non-linear reactive elements. This type of formulation procedure is discussed in Ref. [4] and will not be considered here.

* This work was supported in part by IBM Thomas J. Watson Research Laboratories, Yorktown Heights, New York, and in part by AFOSR Grant AF 1217-67 from the Applied Mathematics Group of the Air Force Office of Scientific Research.

** See also other papers in this conference.

In most switching applications, an initial value for the circuit variables must be generated, which is usually solved by a damped Newton-Raphson or by direct numerical integration. Again, we will not consider the associated numerical problems in order to focus on the usually more demanding transient analysis problem.

IMPLICIT INTEGRATION

To avoid numerical instability problems [4], it is well known that implicit integration procedures must be adopted. In general, these procedures are of one of two forms:

single step (Rosenbrock [5]):

$$\begin{aligned}\underline{1}_r &= [\underline{I} + a_1 T \underline{J}(\underline{x}_r)]^{-1} T \underline{f}(\underline{x}_r) \\ \underline{m}_r &= [\underline{I} + a_2 T \underline{J}(\underline{x}_r)]^{-1} T \underline{f}(\underline{x}_r + b_1 \underline{1}_r) \\ \underline{n}_r &= [\underline{I} + a_3 T \underline{J}(\underline{x}_r)]^{-1} T \underline{f}(\underline{x}_r + b_2 \underline{1}_r + b_3 \underline{m}_r) \\ &\vdots \\ &\vdots \\ &\text{etc.}\end{aligned}$$

$$\underline{x}_{r+1} = \underline{x}_r + R_1 \underline{1}_r + R_2 \underline{m}_r + R_3 \underline{n}_r + \dots \quad (4)$$

where $\underline{J}$ is the Jacobian $\partial \underline{f} / \partial \underline{x}$.

multi-step (Gear [2]):

$$\begin{aligned}\underline{x}_{r+1} &= [\underline{I} + a_0 T \underline{J}(\underline{x}_r)]^{-1} [a_1 \underline{x}_r + \dots + a_{p-r-p+1} \underline{x}_{p-r-p+1} \\ &\quad + T(b_1 \underline{f}_r + b_2 \underline{f}_{r-1} + \dots + b_{p-r-p+1} \underline{f}_{p-r-p+1})] \quad (5)\end{aligned}$$

The numerical stability of both of these classes of methods has been studied elsewhere* [1], [2].

THE DIMENSIONALITY PROBLEM

The dimensionality problem associated with transient analysis of large switching circuits can be recognized at once by observing their structure. In general, a transistor logic element with its interconnections is modeled by a local two-dimensional element interconnection; the

*See paper by Pottle and Allen at this meeting.

associated constant matrices (especially $\tilde{A}$) of Equation (1) therefore tend to appear partitioned (see Figure 1).

We propose this example as motivation only, to demonstrate the relatively low density of matrices. We are in fact interested in adopting methods for handling general forms of sparse matrices which do not have any special block or band structure. We appreciate that by relying exclusively on a general reduction algorithm rather than a mixture of engineering observation (as we have just made) and algorithm we are likely sacrificing considerable computational efficiency.

To conserve on memory and to eliminate time processing zero elements, only the non-zero elements of all matrices should be processed. In general, two techniques may be used:

- (1) the bit-map approach, wherein a word is processed as a series of bits which represent positions in the full matrix; if a bit is "1" it implies a non-zero element is present; this procedure reduces full matrix storage by 32 (IBM 360 series) but is not appropriate for very large matrices;
- (2) the threaded list approach [10], [11] where the non-zero elements are stored in vector form in overall row (column) order, with each row (column) stored in column (row) order. A separate fixed point vector carries the column (row) order. Finally, a fixed point array contains numbers which point to the first position of each row (column) in the first two arrays. For example, the 3×3 matrix

$$A = \begin{bmatrix} 1.1 & 0 & 6.1 \\ 0 & 0 & 7.8 \\ -2 & 3.5 & 0 \end{bmatrix}$$

is stored in column order as

$$A_S = \begin{bmatrix} 1.1 \\ -2.0 \\ 3.5 \\ 6.1 \\ 7.8 \end{bmatrix} \quad N_R = \begin{bmatrix} 1 \\ 3 \\ 3 \\ 1 \\ 2 \end{bmatrix} \quad N_C = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$$

The product of two sparse matrices may be developed in the same row (column) order in which the two original matrices are stored. The algorithm is straightforward and will not be discussed in this paper.

THE EXPERIMENT

We propose to demonstrate by example the savings in computation time possible by use of recently developed integration [2] and sparse matrix algorithms [9]. In particular, the integration methods used are defined as follows:

(1) Euler ($a_1 = -1$, $R_1 = 1$ in Equation (4)); double steps were taken and Richardson extrapolation [6] used to estimate the error;

(2) trapezoidal implicit ($a_1 = -1/2$, $R_1 = 1$); error control method of (1) is used;

(3) third order implicit ($a_1 = a_2 = 0.78866$, $b_1 = -1.1547$, $R_1 = 0.75$, $R_2 = 0.25$); Jacobian evaluation is required at each step [1]; error control method of (1) used;

Before identifying the matrix techniques involved in the experiment, a discussion of certain efficiencies possible because of the form of the network equations is presented. This will motivate our choice of methods for comparison.

All of the implicit procedures of interest involve a reduction of the form $[I + a_1 T J]^{-1} \underline{v}$ where $\underline{v}$ is a vector. The Jacobian, from Equations (1-3), is readily developed as

$$\frac{\partial f}{\partial \underline{x}} = \underline{S}^{-1} \{ \underline{A} + \underline{B} \frac{\partial w}{\partial \underline{u}_n} [I - \underline{F}' \frac{\partial w}{\partial \underline{u}_n}]^{-1} \underline{E} \} + \frac{d(\underline{S}^{-1})f}{d\underline{x}} \quad (5)$$

We assume, as in Reference [12], that the second term is negligible when low accuracy is acceptable. We may then write

$$[I + a_1 T J]^{-1} \underline{v} = \{ \underline{S} + a_1 T (\underline{A} + \underline{B}' \frac{\partial w}{\partial \underline{u}_n} [I - \underline{F}' \frac{\partial w}{\partial \underline{u}_n}]^{-1} \underline{E}) \}^{-1} \underline{S} \underline{v} \quad (6)$$

This preserves the sparsity of the { } term [12]. But Equation (6) is the solution $\underline{y}_2$ of the equation*

$$\begin{bmatrix} -\underline{I} + \underline{F} \frac{\partial \underline{w}}{\partial \underline{u}_n} & \underline{a}_i^T \underline{E} \\ \underline{B}' & \underline{S} + \underline{a}_i \underline{TA} \end{bmatrix} \begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{Sv} \end{bmatrix} \tag{7}$$

The left-hand side matrix will be designated as $\underline{R}$.

The repeated solution of Equation (7) involves either (1) the recalculation of both the L U product form and the vector $\underline{y}_2$ or (2) the recalculation of only $\underline{y}_2$ (as in integration method (3)). The latter operation is trivial, so we will concentrate on the repeated calculation of the product form.

There is obvious profit in performing the logic of the reduction procedure only once, and merely substituting the new values whenever $\underline{R}$ is updated. Several ways of storing this logic have been suggested [9], [11]; we choose the pre-compiled program approach because of its speed advantage in the actual transient solution phase.

There are several other efficiencies worth mentioning in solution for $\underline{y}_2$. First, it is efficient to store the matrix

$$\begin{bmatrix} \underline{F}' \\ \underline{B}' \end{bmatrix}$$

as a single matrix in sparse column order; this composite then premultiplies $\partial \underline{w} / \partial \underline{u}_n$ at each step. Also, updating the matrix $\underline{S} + \underline{a}_i \underline{TA}$ is somewhat difficult in sparse form, since it seemingly involves "re-threading" the column order list when $\underline{S}$ is changed. It is more efficient to leave appropriate holes in the composite threaded list. The positions of these holes are then stored and the new values for $\underline{S}$ added and subtracted at each update.

The matrix methods involved in reduction of $\underline{R}$ are:

* A similar equation arises in the Newton-Raphson solution of the D. C. analysis problem.

(A) full matrix operations, without L U factorization of R;
 (B) sparse matrix operations, with L U factorization, without saving information about sparse structure; only non-zero entries are processed, in contrast to (A);

(C) sparse matrix operations, with L U factorization and storage; in this case, a special subroutine is compiled using the Crout algorithm (without pivoting) which sequentially processes the required reduction operations without loops and without explicit table-look-up.*

THE CIRCUIT

A current switch and associated emitter follower is to serve as an example for comparison purposes (Figure 2). The densities of the associated matrices are

$$\begin{array}{lll} \underline{A}(32\%) & , & \underline{B}(4.9\%) & , & \underline{B}'(11\%) \\ \underline{E}(9\%) & , & \underline{F}(0\%) & , & \underline{F}'(0\%) \\ \underline{S}(8.8\%) & , & \underline{R}(22\%) \end{array}$$

The densities are quite high because the sparse transmission line coupling matrices are not included.

CONCLUSIONS

In this compacted paper full discussion of the results is impossible. Only three points will be made:

1) The use of a pre-compiled matrix reduction program is essential for evaluating the worth of integration strategies. One can readily show the profit in higher order integration methods if the pre-compiled program is not used; with the latter, this is not nearly so obvious (Table I(b), (g), (h)).

2) Explicit integration methods have no particular advantage over implicit, provided no extravagant fill-in occurs in reduction of $\underline{R}$.

3) The next level of sophistication is the use of pre-compiled multiplication techniques. An excessive amount of time is evident in the iterative solution of Equation (3) and in the updating of the right-hand side

* This program was developed by Dr. Fred Gustavson of IBM, Yorktown, [8].

of Equation (7)

ACKNOWLEDGEMENT

The author expresses his appreciation to Drs. Fred Gustavson, Gary Hachtel, and Robert Brayton of IBM Yorktown and to Dr. F. H. Branin, Jr., of IBM Kingston for discussions of the numerical state-of-the-art in this area of analysis. The invaluable assistance of T. E. Grapes in programming these experiments is also acknowledged.

REFERENCES

- [1]. D. A. Calahan, "A Stable, Accurate Method of Numerical Integration for Nonlinear Systems," Proc IEEE, vol. 56, p744; April, 1968.
- [2]. C. W. Gear, "Numerical Integration of Stiff Ordinary Differential Equations," Report No. 221, Department of Computer Science, University of Illinois; January 20, 1967.
- [3]. O. B. Widlund, "A Note on Unconditionally Stable Linear Multistep Methods," BIT, vol. 7, pp 65-70; 1967.
- [4]. D. A. Calahan, Computer-Aided Network Design (Preliminary Edition); McGraw-Hill, January 1968.
- [5]. H. H. Rosenbrock, "Some General Implicit Processes for the Numerical Solution of Differential Equations," Computer Journal, vol. 12, pp 329-330; 1964.
- [6]. A. Ralston, A First Course in Numerical Analysis; McGraw-Hill, 1965.
- [7]. F. H. Branin, Jr., "Sparse Matrices in Network Analysis," Digest of the Symposium on Sparse Matrices, (IBM Yorktown); September 9-10, 1968.
- [8]. R. Brayton, et al, "A Comparison Between EFI and PRI for Sparse Matrices," Digest of the Symposium on Sparse Matrices; (IBM Yorktown); September 9-10, 1968.
- [9]. F. Gustavson, et al, "Symbolic Generation of An Optimal Crout Algorithm for Sparse Systems of Linear Equation," IBM Report RC 1852 (Yorktown Heights Laboratory); June 27, 1967.
- [10]. D. M. Smith, "Data Logistics for Matrix Inversion," Digest of the Sparse Matrix Symposium, (IBM Yorktown); September 9-10, 1968.
- [11]. H. B. Lee, "An Interpretive Implimentation of Gaussian Elimination," Digest of the Sparse Matrix Symposium, (IBM Yorktown); September 9-10, 1968.
- [12]. G. D. Hachtel, and R. A. Rohrer, "Techniques for the Optimal Design and Synthesis of Switching Circuits," Proc. IEEE, vol. 55, pp 1864-1876; November, 1967.

<u>Matrix method</u>	<u>Total time</u>	<u>Integration Method</u>	<u>No. of Steps</u>	<u>Total Times</u>
Full	102. s			
Sparse	22.5 s	Euler	133	10.76 s
Pre-compiled		Trap.	57	3.71 s
sparse	1.97s	Third Order	35	4.33 s

(a) Trapezoidal integration;
absolute error=.01; 24 steps.

(b) precompiled sparse matrix
reduction; absolute error=.001

<u>Absolute Error</u>	<u>No. of Steps</u>	<u>Total Time</u>
.01	24	1.97 s
.001	57	3.71 s
.0001	166	12.32 s

(c) Trapezoidal integration;
precompiled sparse reduction

(d) Other Data

- 1) Experiments run on IBM/360-67 duplex time-sharing system.
- 2) Matrices generated by Gustavson's program in 3.29 sec.
- 3) R has 183 entries; its LU product form has 253 entries; S is responsible for 6 new entries in R and none in its LU product.
- 4) Largest/smallest eigenvalue= 4.3×10^8

(e) Euler and trapezoidal;
sparse

(f) Euler and trapezoidal;
precompiled sparse

	<u>Time (msec)</u>
algebraic solution	20
total update time (Table I(f))	26
reduce R	440
reform (Table I(f))	11
reduce R	440
integration logic	2
	<u>939</u>

error control: 48.3%

	<u>Time (msec)</u>
algebraic solution	10
rhs of Eq. 7	9
update S	3
left side of R	8
right side of R (with new step size)	6
reduce Eq. 7	13
integration logic;	
right side of R	11
reduce R	13
integration logic	2
	<u>75</u>

error control: 31.6%

Table I. Summary of Experimental Data;

(parts (e) -(j) give typical
computing times per integration step).

(g) Third order; sparse

	Time (msec)
First 939 msec	
same as Table I(e)	939*
(2 LU products saved)	
algebraic solution (2)	40*
total update time (2)	52*
solve stored matrix (2)	100*
integration logic (2)	4*
	1135
error control: 48.5%	
*These values are estimated from other data.	

(h) Third order; precompiled sparse

	Time (msec)
First 75 msec	
same as Table I(f)	75
(2 LU products saved)	
algebraic solution (2)	20
rhs of Eq. 7 (2)	18
solve stored matrices (2)	7
integration logic (2)	4
	124
error control: 42.7%	

(i) Second-order Runge-Kutta;
sparse multiplication

	Time (msec)
algebraic solution (3)	30
(precompiled matrix reduction)	
derivative update (3)	27
update S	
form $S^{-1}f$ (precompiled)	3
integration logic	2
	62
error control: 32.2%	

(j) Trapezoidal; full matrix methods

	Time (msec)
algebraic solution	160
rhs of Eq. 7 (2)	180
matrix reduction (2)	3900
	4240
error control: 48.2%	

Table I (continued).

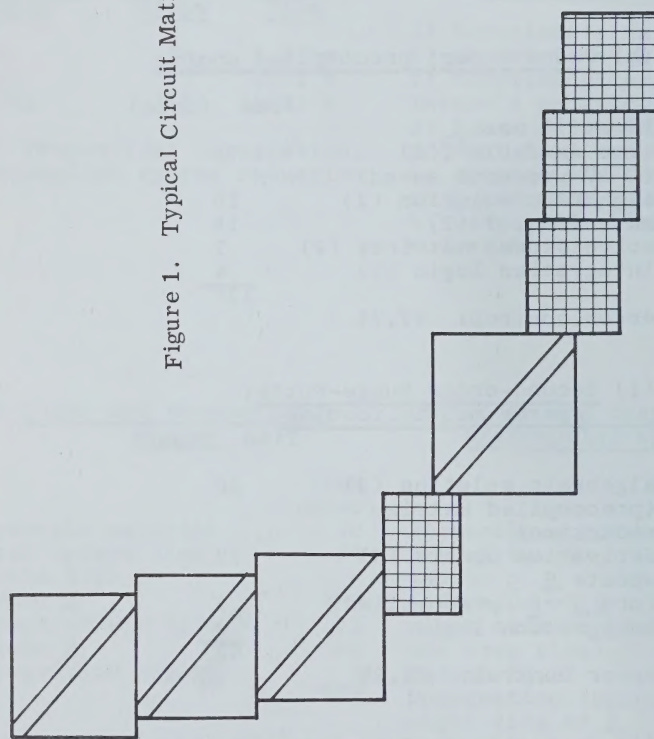


Figure 1. Typical Circuit Matrix Structure



20% DENSE MATRIX DESCRIPTION OF TRANSISTOR



UPPER AND LOWER DIAGONAL MATRIX DESCRIBING
TRANSMISSION LINE

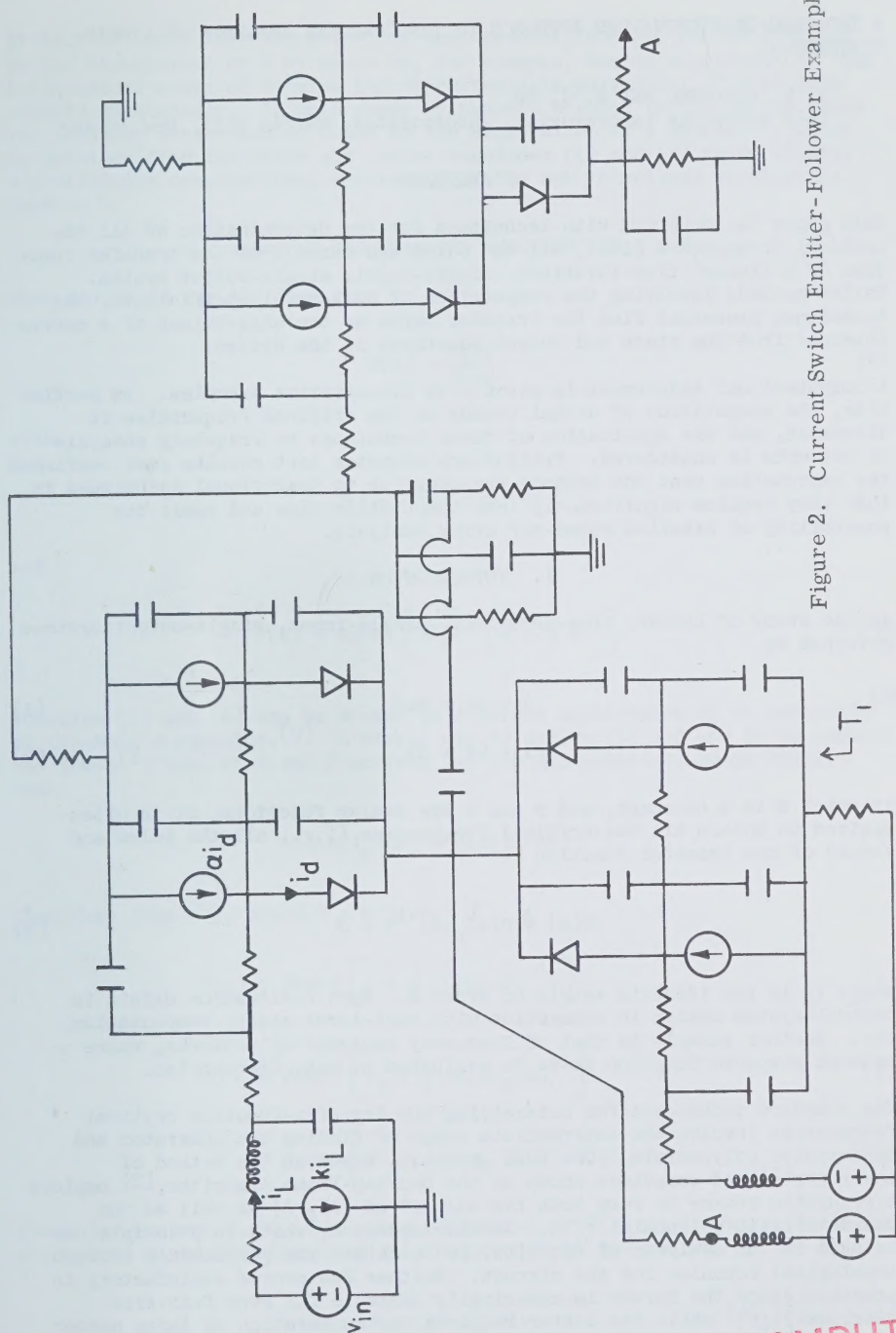


Figure 2. Current Switch Emmitter-Follower Example

A TWO-SETS-OF-EIGENVALUES APPROACH TO THE COMPUTER ANALYSIS OF LINEAR SYSTEMS

I. W. SANDBERG AND H. C. SO

Bell Telephone Laboratories, Incorporated; Murray Hill, New Jersey

ABSTRACT

This paper is concerned with techniques for the determination of all the critical frequencies (i.e., all the poles and zeros), of the transfer function of a linear, time-invariant, single-input, single-output system. Unlike methods involving the computation of polynomial coefficients, the techniques presented find the transfer zeros as the eigenvalues of a matrix obtained from the state and output equations of the system.

A computational assessment is given with illustrative examples. In particular, the computation of actual bounds on the critical frequencies is discussed, and the application of these techniques to frequency analysis of networks is considered. Preliminary computer test results have confirmed the expectation that the methods are superior to traditional techniques in that they require significantly less computation time and admit the possibility of detailed round-off error analysis.

1. INTRODUCTION

In the study of linear, time-invariant, single-input, single-output systems governed by

$$\dot{x} = Ax + Bu \quad (1)$$

$$y = Cx + du \quad (2)$$

in which d is a constant, and y and u are scalar functions, it is often desired to obtain all the critical frequencies (i.e., all the poles and zeros) of the transfer function

$$T(s) \triangleq C(sI_N - A)^{-1}B + d \quad (3)$$

where I_N is the identity matrix of order N . Such a situation exists in control-system design in connection with root-locus study, compensation, etc. Another example is that of frequency analysis of networks, where a network response function is to be evaluated at many frequencies.

The standard techniques for determining the transfer-function critical frequencies involve the intermediate stage of finding the numerator and denominator polynomials. One such approach, based on the method of Leverrier^[1] and sometimes known as the Souriau-Frame Algorithm,^[2] employs a recursive scheme to form both the adjoint of $[sI_N - A]$ as well as the characteristic polynomial of A . Another approach, which in principle can be used in the analysis of circuits, is to obtain the polynomials through topological formulae for the circuit. Neither has proved satisfactory in practice since the former is numerically unstable for even fair-size matrices^{[3],[4]} while the latter requires the enumeration of large number of trees.^[5]

It is well known that the poles¹ of the transfer function can be computed as the eigenvalues of A by means of, for example, the QR algorithm^[7]. The intermediate stage of forming the characteristic polynomial of A is thus clearly unnecessary. In this paper we present several techniques by which the transfer zeros¹ can be found as the eigenvalues of a matrix A^* , which is obtained from the state and output equations (1) and (2) quite simply. A preliminary computational assessment of the new techniques is given in Section 6.

2. ZEROS OF THE TRANSFER FUNCTION FOR $d \neq 0$

When $d \neq 0$, we use the fact that

$$T(s) = \frac{P(s)}{Q(s)} \quad (4)$$

with

$$Q(s) = \det[sI_N - A], \quad (5)$$

and

$$P(s) = d \det \left[sI_N - \left(A - \frac{1}{d} BC \right) \right]. \quad (6)$$

Formulae (5) and (6) can be viewed as a direct application of an extension^[8] of Blackman's equation.^[9] A simple way to derive (5) and (6) is to exploit the lemma^[8] that if E and F are any $I \times N$ and $N \times L$ matrices, respectively, then

$$\det[I_L + EF] = \det[I_N + FE]. \quad (7)$$

Therefore from (3), since $d \neq 0$ here,

$$T(s) = d \det \left[I + \frac{1}{d} C(sI_N - A)^{-1} B \right] \quad (8)$$

$$= d \det \left[I_N + \frac{1}{d} BC(sI_N - A)^{-1} \right] \quad (9)$$

$$= d \det \left[sI_N - A + \frac{1}{d} BC \right] / \det[sI_N - A]. \quad (10)$$

Thus, for nonzero d, the numerator polynomial of $T(s)$ is of degree N and is given by d times the characteristic polynomial of the matrix

$$A^* = A - \frac{1}{d} BC. \quad (11)$$

Clearly, the zeros of $T(s)$ can be obtained as the eigenvalues of A^* defined in (11).

3. ITERATIVE REDUCTION TECHNIQUE FOR $d = 0$

When $d = 0$, we have

$$T(s) = C(sI_N - A)^{-1}B. \quad (12)$$

Using a well-known determinantal identity^[10], equation (3) can be written as

$$T(s) = \det \begin{bmatrix} d & -C \\ B & (sI_N - A) \end{bmatrix} / \det[sI_N - A]. \quad (13)$$

Hence for $d = 0$, we have, for $T(s) = P(s)/Q(s)$,

$$Q(s) = \det[sI_N - A], \quad (14)$$

$$P(s) = \det \begin{bmatrix} 0 & -C \\ B & (sI_N - A) \end{bmatrix}. \quad (15)$$

It is clear that when d vanishes $P(s)$ is of lower degree than N . We are therefore seeking to put the numerator polynomial into the form

$$P(s) = k \det[sI_W - A^*] \quad (16)$$

where W is the degree of $P(s)$, A^* is a $W \times W$ matrix, and k is a constant.² In the remainder of this section we first introduce some definitions and notation; then we present a theorem which, when applied to equation (15), gives a basis for generating a suitable matrix A^* for (16); finally we describe an algorithm for generating A^* .

Notation and Definitions

For any square matrix G of order at least ℓ , we shall use $(\ell)G$ and $G(\ell)$ to denote the ℓ^{th} row and column, respectively, of G ; $\bar{\ell}G$ or $G_{\bar{\ell}}$ denotes the matrix resulting from the deletion of the ℓ^{th} row or column, respectively, from G .

Definition 1: $V_M^{(\ell)}$ is a row matrix $[v_1, v_2, \dots, v_M]$ in which $v_i = 0$ if $i \neq \ell$ and $v_\ell = 1$.

Definition 2: $\psi_M^{(\ell)}$ is the transpose of $V_M^{(\ell)}$.

$$T_M^{(\ell)} = \begin{bmatrix} 1 & & & & h_1/h_\ell & & & \\ & 1 & & & h_2/h_\ell & & & \\ & & \ddots & & \vdots & & & \\ & & & 1 & h_{\ell-1}/h_\ell & & & \\ & & & & 1 & & & \\ & & & & h_{\ell+1}/h_\ell & 1 & & \\ & & & & \vdots & & \ddots & \\ & & & & h_M/h_\ell & & & 1 \end{bmatrix} \quad (20)$$

Proof: We have

$$\det \begin{bmatrix} O & -J \\ H & sI_M - G \end{bmatrix} = \det \begin{bmatrix} h_\ell & -J - (\ell)^{G+sV_M^{(\ell)}} \\ H & sI_M - G \end{bmatrix} \quad (21)$$

$$= h_\ell \times \det \left[[sI_M - G] + \frac{1}{h_\ell} H [J + (\ell)^G - sV_M^{(\ell)}] \right] \quad (22)$$

$$= h_\ell \times \det \left[s \left[I_M - \frac{1}{h_\ell} HV_M^{(\ell)} \right] + \left[\frac{1}{h_\ell} H(J + (\ell)^G) - G \right] \right] \quad (23)$$

$$= h_\ell \times \det \left[sE_M^{(\ell)} \left[I_M - \frac{1}{h_\ell} HV_M^{(\ell)} \right] T_M^{(\ell)} E_M^{(\ell)} + E_M^{(\ell)} \left[\frac{1}{h_\ell} H(J + (\ell)^G) - G \right] T_M^{(\ell)} E_M^{(\ell)} \right] \quad (24)$$

Note that

$$\left[I_M - \frac{1}{h_\ell} HV_M^{(\ell)} \right] T_M^{(\ell)} = [\rho_{ij}] \quad (25)$$

where

$$\begin{aligned}\rho_{ij} &= 0 & i \neq j \\ \rho_{ii} &= 1 & i \neq l \\ \rho_{ll} &= 0\end{aligned}\tag{26}$$

Therefore

$$\det \begin{bmatrix} 0 & -J \\ H & sI_M - G \end{bmatrix} = h_l \times \det \left\{ \begin{bmatrix} 0 & 0 \\ 0 & sI_M - l \end{bmatrix} + E_M^{(l)} \left[\frac{1}{h_l} H(J^{+}_{(l)}G) - G \right] T_M^{(l)} E_M^{(l)} \right\}.\tag{27}$$

To show that $\delta = \frac{1}{h_l} JH$, observe that the l^{th} row of $\left[\frac{1}{h_l} H(J^{+}_{(l)}G) - G \right]$ is simply J . Thus the l, l element of $\left[\frac{1}{h_l} H(J^{+}_{(l)}G) - G \right] T_M^{(l)}$, which is δ , is indeed given by $J[h_1/h_l, h_2/h_l, \dots, h_{l-1}/h_l, 1, h_{l+1}/h_l, \dots, h_M/h_l]^t$. $\square$

Remarks: We could write

$$\det \begin{bmatrix} 0 & -J \\ H & sI_M - G \end{bmatrix} = \det \begin{bmatrix} 0 & H^t \\ -J^t & sI_M - G^t \end{bmatrix} = \det \begin{bmatrix} 0 & -J' \\ H' & sI_M - G' \end{bmatrix}\tag{28}$$

and then apply Theorem 1 to the right-hand side. The result would be "dual" to (27) as a nonzero element in J , rather than in H , would be used to effect the matrix reduction.

In view of Equation (15) and Theorem 1, the following iterative algorithm

can be applied to $\begin{bmatrix} 0 & -C \\ B & sI_N - A \end{bmatrix}$ to determine k and a suitable matrix A^*

for (16).

- (a) Search the first column of the matrix under consideration for a nonzero entry. If no such nonzero entry is found, $T(s)$ is identically zero.

- (b) Use Equations (18) and (19) to replace the present matrix with one that is smaller in order by 1. The transformations represented by $T_M^{(\ell)}$ and $E_M^{(\ell)}$ should be effected directly to avoid matrix product computations.
- (c) Test δ . If δ is zero repeat steps (a) and (b). If δ is not zero the results of the last section apply.
- (d) The reduction is repeated at most N times since the matrix $\begin{bmatrix} 0 & -C \\ B & sI_N - A \end{bmatrix}$ reduced N times is a constant which is nonzero if $T(s) \neq 0$.

4. EXPLICIT MATRIX FORMULATION: GENERAL

As an alternative to the iterative technique of the last section, we now describe an approach in which the transfer function is expressed as the ratio of two explicitly defined determinants when $d = 0$. We begin with

$$T(s) = \frac{1}{s} C \left[I_N - \frac{1}{s} A \right]^{-1} B \quad (29)$$

For sufficiently large $|s|$ (e.g., for $\| \frac{1}{s} A \| < 1$),^[11]

$$T(s) = \frac{1}{s} C [I_N + A/s + A^2/s^2 + \dots] B. \quad (30)$$

It is easy to show that (30) implies that for all $|s|$ such that $\| \frac{1}{s} A \| < 1$,

$$T(s) = \frac{1}{s} [CB + CAB/s + CA^2B/s^2 + \dots] \quad (31)$$

From (31) it is clear that if

$$CA^p B \neq 0 \quad p \text{ a non-negative integer} \quad (32)$$

$$CA^j B = 0 \quad \text{for all non-negative integers } j < p \quad (33)$$

then the polynomial $P(s)$ of $T(s)$ is of degree $N-p-1$. It is also clear that (32) and (33) hold for some $p \leq N-1$, and the leading coefficient of $P(s)$ is $CA^p B$.

Theorem 2: Let p be a non-negative integer such that

$$CA^p B \neq 0$$

and $CA^j B = 0$ for all integers $j, 0 \leq j < p$.

Let ℓ be any positive integer such that the ℓ^{th} element of B is nonzero (i.e., let $b_\ell \neq 0$). Then³

$$T(s) = \frac{k}{s^p} \frac{\det \left[I_{N-1} s - \left(I_N - \frac{1}{b_l} B V_N^{(l)} \right) A \left(I_N - \frac{B C A^p}{k} \right) \bar{l} \right]}{\det [I_N s - A]} \quad (34)$$

where $k = C A^p B$.

Proof: Let $|s|$ be such that $\| \frac{1}{s} A \| < 1$. Then by the hypothesis and (31),

$$T(s) = \frac{1}{s} \left[\frac{C A^p B}{s^p} + \frac{C A^{p+1} B}{s^{p+1}} + \dots \right] \quad (35)$$

$$= \frac{1}{s^{p+1}} C A^p B + \frac{1}{s} \left[\frac{C A^{p+1} B}{s^{p+1}} + \frac{C A^{p+2} B}{s^{p+2}} + \dots \right] \quad (36)$$

$$= \frac{1}{s^{p+1}} C A^p B + \frac{1}{s^{p+2}} C A^{p+1} \left[I_N + \frac{A}{s} + \frac{A^2}{s^2} + \dots \right] B. \quad (37)$$

Thus, with $k = C A^p B$,

$$T(s) = \frac{k}{s^{p+1}} \det \left[1 + \frac{1}{k s} C A^{p+1} \left(I_N - \frac{1}{s} A \right)^{-1} B \right]. \quad (38)$$

By identity (7),

$$T(s) = \frac{k}{s^{p+1}} \det \left[I_N + \frac{1}{k s} B C A^{p+1} \left(I_N - \frac{1}{s} A \right)^{-1} \right] \quad (39)$$

$$= \frac{k}{s^{p+1}} \frac{\det \left[\left(I_N - \frac{1}{s} A \right) + \frac{1}{k s} B C A^{p+1} \right]}{\det \left[I_N - \frac{1}{s} A \right]} \quad (40)$$

$$T(s) = \frac{k}{s^{p+1}} \frac{\det \left[s I_N - A + \frac{1}{k} B C A^{p+1} \right]}{\det [s I_N - A]}. \quad (41)$$

Since Equation (41) holds for all s such that $\| \frac{1}{s} A \| < 1$, and since both sides of (41) are real rational functions of s , it follows that (41) holds for all s .⁴

Suppose now that $b_\ell \neq 0$. Let $L^{(\ell)}$ denote the $N \times N$ matrix

$$L^{(\ell)} = \begin{bmatrix} 1 & & & & b_1 & & & \\ & 1 & & & b_2 & & & \\ & & \ddots & & \vdots & & & \\ & & & 1 & b_{\ell-1} & & & \\ & & & & b_\ell & & & \\ & & & & b_{\ell+1} & 1 & & \\ & & & & \vdots & & \ddots & \\ & & & & b_N & & & 1 \end{bmatrix} \quad (42)$$

Then

$$L^{(\ell)} = \left[I_N + \left(B - \psi_N^{(\ell)} \right) V_N^{(\ell)} \right], \quad (43)$$

and

$$[L^{(\ell)}]^{-1} = \begin{bmatrix} 1 & & & & -b_1/b_\ell & & & \\ & 1 & & & -b_2/b_\ell & & & \\ & & \ddots & & \vdots & & & \\ & & & 1 & -b_{\ell-1}/b_\ell & & & \\ & & & & 1/b_\ell & & & \\ & & & & -b_{\ell+1}/b_\ell & 1 & & \\ & & & & \vdots & & \ddots & \\ & & & & -b_N/b_\ell & & & 1 \end{bmatrix}. \quad (44)$$

Let $\tau = \left[I_N - \frac{1}{k} BCA^p \right]$. Since $\tau B = 0$, the product

$$\tau' = \tau L^{(\ell)} = \tau \left[I_N + \left(B - \psi_N^{(\ell)} \right) V_N^{(\ell)} \right] \quad (45)$$

$$= \tau \left[I_N - \psi_N^{(\ell)} V_N^{(\ell)} \right] \quad (46)$$

is identical to τ except that the ℓ^{th} column has all zeros. Therefore

$$\tau = \tau' [L^{(\ell)}]^{-1} \quad (47)$$

$$= \tau_{\bar{\ell}} \times_{\bar{\ell}} [L^{(\ell)}]^{-1} \quad (48)$$

where

$$_{\bar{\ell}} [L^{(\ell)}]^{-1} = _{\bar{\ell}} \left[I_N - \frac{1}{b_{\ell}} B V_N^{(\ell)} \right]. \quad (49)$$

Thus

$$\tau = \left[I_N - \frac{1}{k} B C A^p \right] = \left[I_N - \frac{1}{k} B C A^p \right]_{\bar{\ell}} \times_{\bar{\ell}} [L^{(\ell)}]^{-1}. \quad (50)$$

Now, from (41)

$$T(s) = \frac{k}{s^{p+1}} \frac{s^N \det \left[I_N - \frac{1}{s} \left(I_N - \frac{1}{k} B C A^p \right) A \right]}{\det[sI_N - A]}. \quad (51)$$

By identity (7),

$$T(s) = \frac{k}{s^{p+1}} \frac{s^N \det \left[I_N - \frac{1}{s} A \left(I_N - \frac{1}{k} B C A^p \right) \right]}{\det[sI_N - A]} \quad (52)$$

$$= \frac{k s^N}{s^{p+1}} \frac{\det \left[I_N - \frac{1}{s} A \left(I_N - \frac{1}{k} B C A^p \right) \right]_{\bar{\ell}} \times_{\bar{\ell}} [L^{(\ell)}]^{-1}}{\det[sI_N - A]} \quad (53)$$

$$= \frac{k}{s^p} \frac{\det \left[sI_{N-1} - _{\bar{\ell}} (L^{(\ell)})^{-1} A \left(I_N - \frac{1}{k} B C A^p \right) \right]_{\bar{\ell}}}{\det[sI_N - A]} \quad (54)$$

where $_{\bar{\ell}} (L^{(\ell)})^{-1}$ is the matrix in (44) with the ℓ^{th} row deleted. $\square$

Remarks: To find the transmission zeros, one might form the matrix

$$A^* = \bar{\ell}^{(L^{(\ell)})^{-1}} A \left(I_N - \frac{1}{k} B C A^p \right)_{\bar{\ell}} \quad (55)$$

and solve for the eigenvalues of A^* . There are p superfluous eigenvalues in A^* ; all these eigenvalues are zero.

Note that in (55) the pre-multiplying matrix $\bar{\ell}^{(L^{(\ell)})^{-1}}$ represents very simple operations on A . Since

$$\bar{\ell} \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & \ddots & & & & & & \\ & & & 1 & & & & & \\ & & & & -b_{\ell-1}/b_{\ell} & & & & \\ & & & & 1/b_{\ell} & & & & \\ & & & & -b_{\ell+1}/b_{\ell} & 1 & & & \\ & & & & & & \ddots & & \\ & & & & & & & & 1 \end{bmatrix} A = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & \ddots & & & & & & \\ & & & 1 & & & & & \\ & & & & -b_{\ell-1}/b_{\ell} & & & & \\ & & & & 1 & & & & \\ & & & & -b_{\ell+1}/b_{\ell} & 1 & & & \\ & & & & & & \ddots & & \\ & & & & & & & & 1 \end{bmatrix} A. \quad (56)$$

The product $\bar{\ell}^{(L^{(\ell)})^{-1}} A$ can be obtained by $(N-1)$ operations of multiplying the ℓ^{th} row of A by a constant and then adding it to another row. The ℓ^{th} row is then deleted.

In general it is necessary to compute CB and CA^jB for $j = 1, 2, \dots$ until a nonzero CA^jB is found. The fact that C is a row matrix is of great help here. Furthermore, there are important special cases, discussed in the next section, for which the application of Theorem 2 is particularly simple.

5. EXPLICIT MATRIX FORMULATION: SPECIAL CASES

(a) If $CB \neq 0$, Equation (34) takes on a particularly simple form for then $p = 0$ and

$$T(s) = CB \frac{\det \left[I_{N-1}s - \left(I_N - \frac{1}{b_\ell} BV_N^{(\ell)} \right) A \left(I_N - \frac{BC}{CB} \right) \right]}{\det[I_N s - A]}. \quad (57)$$

For example, it is well known that, for any positive-element network of linear resistors, capacitors, inductors and transformers (RLCM), the numerator and the denominator of a driving-point function do not differ in degree by more than 1. Thus if $T(s)$ is a driving-point function of any such network, and if $d = 0$, then p must be zero; that is, (57) applies.

(b) If $B = [b_1, b_2, \dots, b_N]^T$ and $C = [c_1, c_2, \dots, c_N]$ are such that

$$\begin{aligned} c_\ell b_\ell &\neq 0 && \text{for some positive integer } \ell \\ c_j b_j &= 0 && \text{for all } j \neq \ell \end{aligned} \quad (58)$$

then we can further simplify (57) to:⁵

$$T(s) = CB \frac{\det \left[sI_{N-1} - \bar{I} A' \right]}{\det[sI_N - A']} \triangleq c_\ell b_\ell \frac{\Delta_{\ell\ell}}{\Delta} \quad (59)$$

where

$$A' = \theta_2 \theta_1^{-1} A \theta_1 \theta_2^{-1} \quad (60)$$

$$\Delta = \det[sI_N - A'] = \det[sI_N - A] \quad (61)$$

$$\Delta_{\ell\ell} = \det_\ell[sI_N - A']_\ell, \quad (62)$$

θ_1 is obtained from the identity matrix by replacing the ℓ^{th} row with

$$\left[\frac{-c_1}{c_\ell}, \frac{-c_2}{c_\ell}, \dots, \frac{-c_{\ell-1}}{c_\ell}, 1, \frac{-c_{\ell+1}}{c_\ell}, \dots, \frac{-c_N}{c_\ell} \right], \text{ and } \theta_2 \text{ is obtained from the}$$

identity matrix by replacing the ℓ^{th} column with $\begin{bmatrix} -b_1 \\ b_\ell \\ \vdots \\ -b_{\ell-1} \\ b_\ell \end{bmatrix}$, $\frac{-b_{\ell+1}}{b_\ell}, \dots, \frac{-b_N}{b_\ell}$. The matrices θ_1 and θ_2 have the property that

$$c\theta_1 = c_\ell v_N^{(\ell)} = [0, \dots, 0, c_\ell, 0, \dots, 0], \quad (63)$$

$$\theta_2^B = b_\ell v_N^{(\ell)} = [0, \dots, 0, b_\ell, 0, \dots, 0]^T. \quad (64)$$

It is interesting to note that whenever $d = 0$ and $p = 0$, then $T(s)$ can be written as

$$CB \frac{\det \tilde{\ell} [sI_N - M] \tilde{\ell}}{\det [sI_N - M]}, \quad (65)$$

since an initial elementary linear transformation of the state and output equations can be employed if necessary in order that (58) is satisfied. In particular, $T(s)$ can be expressed in the form (65) whenever $T(s)$ is a (nonidentically zero) driving-point function, with a zero at infinity, of a passive network.

Example 1: Consider the network in Fig. 1. The differential equations are

$$\dot{x} = \begin{bmatrix} \dot{v}_{C_1} \\ \dot{v}_{C_2} \\ \dot{v}_{C_3} \\ \dot{i}_{L_1} \\ \dot{i}_{L_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{C_1} & 0 \\ 0 & 0 & 0 & \frac{1}{C_2} & -\frac{1}{C_2} \\ 0 & 0 & -\frac{G_1}{C_3} & 0 & \frac{1}{C_3} \\ \frac{1}{L_1} & -\frac{1}{L_1} & 0 & 0 & 0 \\ 0 & \frac{1}{L_2} & -\frac{1}{L_2} & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{C_1} \\ v_{C_2} \\ v_{C_3} \\ i_{L_1} \\ i_{L_2} \end{bmatrix} + \begin{bmatrix} \frac{1}{C_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} j_s(t) \quad (66)$$

The output equation for the driving-point case is

$$v_s(t) = [1, 0, 0, 0, 0]x$$

Here θ_1 and θ_2 are both the identity matrix; $b_k = b_1 = \frac{1}{C_1}$, and $c_k = c_1 = 1$. For this example,

$$T(s) = Z_{in}(s) = \frac{1}{C_1} \frac{\det[sI_{N-1} - \bar{1}A_1]}{\det[sI_N - A]}$$

$$= \frac{1}{C_1} \frac{\Delta_{11}}{\Delta}$$

where

$$\Delta = \begin{vmatrix} s & 0 & 0 & \frac{1}{C_1} & 0 \\ 0 & s & 0 & -\frac{1}{C_2} & \frac{1}{C_2} \\ 0 & 0 & s + \frac{G_1}{C_3} & 0 & -\frac{1}{C_3} \\ -\frac{1}{L_1} & \frac{1}{L_1} & 0 & s & 0 \\ 0 & -\frac{1}{L_2} & \frac{1}{L_2} & 0 & s \end{vmatrix}.$$

For $C_1 = \frac{1}{10}$, $C_2 = \frac{1}{4}$, $C_3 = \frac{1}{8}$, $L_1 = 1$, $L_2 = \frac{1}{2}$, $R_1 = 1$:

$$A = \begin{bmatrix} 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 4 & -4 \\ 0 & 0 & -8 & 0 & 8 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 \end{bmatrix}.$$

The eigenvalues of A (i.e., the poles of Z_{in}) are given below with bounds⁶ on the magnitudes of the errors:

$$\begin{aligned}
 & - 0.2931800 \pm j4.178528 + \epsilon_1, \quad - 0.8509527 \pm j2.379481 + \epsilon_2, \\
 & - 5.711735 + \epsilon_3;
 \end{aligned}$$

$$|\epsilon_1| \leq 0.6241800 \times 10^{-7}, \quad |\epsilon_2| \leq 0.3765619 \times 10^{-7},$$

$$|\epsilon_3| \leq 0.8511148 \times 10^{-7}.$$

The eigenvalues of $\bar{1}A_1$ (zeros of Z_{in}) are:

$$\begin{aligned}
 & - 0.7404065 \pm j3.621251 + \epsilon_4, \quad - 0.8223190 + \epsilon_5, \\
 & - 5.696868 + \epsilon_6;
 \end{aligned}$$

$$|\epsilon_4| \leq 0.5507720 \times 10^{-7}, \quad |\epsilon_5| \leq 0.1225351 \times 10^{-7},$$

$$|\epsilon_6| \leq 0.8488995 \times 10^{-7}.$$

For the transfer function $\frac{V_{out}(s)}{J(s)}$, the output equation is

$$v_{out} = [0, 0, 1, 0, 0]x$$

Thus

$$CB = 0$$

$$CA = [0, 0, -8, 0, 8]$$

$$CA^2 = [0, 16, 48, 0, -64]$$

$$CA^3 = [0, -128, -256, 64, 320]$$

$$CA^4 = [64, 576, 1408, -512, -1536]$$

$$CB = CAB = CA^2B = CA^3B = 0$$

$$CA^4B = 640.$$

Using Theorem 2, Equation (34), for element values given above,

$$T(s) = \frac{640}{s^4} \frac{\det \left[\begin{array}{c} I_{N-1}s - \left(I_N - \psi_N^{(1)} V_N^{(1)} \right) A \left(I_N - \frac{1}{64} \psi_N^{(1)} C A^4 \right) \\ \hline \end{array} \right]}{\det[sI_N - A]}$$

$$= \frac{640}{s^4} \frac{\det \left[\begin{array}{c} I_{N-1}s - \begin{pmatrix} 0 & 0 & 0 & 4 & -4 \\ 0 & 0 & -8 & 0 & 8 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 \end{pmatrix} \begin{pmatrix} -9 & -22 & 8 & 24 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ \hline \end{array} \right]}{\det[sI_N - A]}$$

$$= \frac{640}{s^4} \frac{\det \left[\begin{array}{c} I_{N-1}s - \begin{pmatrix} 0 & 0 & 4 & -4 \\ 0 & -8 & 0 & 8 \\ -10 & -22 & 8 & 24 \\ 2 & -2 & 0 & 0 \end{pmatrix} \\ \hline \end{array} \right]}{\det[sI_N - A]}$$

$$T(s) = 640 \frac{1}{\det[sI_N - A]} .$$

The last expression could have been obtained more directly: since $p = 4 = N - 1$, the numerator is a constant given by $CA^4B = 640$.

6. PRELIMINARY COMPUTATIONAL ASSESSMENT

The computational techniques described in this paper have been tested on a GE645 digital computer at Bell Laboratories. The eigenvalues of the "numerator and denominator matrices" are calculated with a computer program which implements the QR algorithm. In the following discussion we first make some general comments on the techniques presented. We then discuss mainly the applications to network analysis, and we give circuit examples.

(a) Computational Aspects of the Methods Presented:

(i) The iterative reduction method of Section 3 is simple and can be executed rapidly on a digital computer. Care must be taken, however, in determining whether δ is zero. Since δ is proportional to the coefficient of the highest power of the numerator polynomial $P(s)$ of $T(s)$, the decision at some cycle of the reduction that δ is not zero is equivalent to deciding the degree of $P(s)$. In general, due to round-off errors in the computation, δ is some small number when it should be zero. Whether or not a small but nonzero computed δ should be treated as zero depends on the scale of numbers in A , B , and C , the number of iterations performed, etc.

(ii) The explicit matrix-formulation method of Section 5 leads to a matrix A^* of Equation (55), which has p superfluous zero eigenvalues. If p is not small, the accuracy with which all the eigenvalues of A^* can be determined by well-known methods appears to be uncertain. This consideration, together with the fact that CA^jB has to be calculated for successive values of j until p is found, seems to suggest that for computation purposes Theorem 2 is best used when p is known to be small.

(iii) We have not considered the effects of round-off errors on the computed "numerator matrix," and hence on the computed zeros of $P(s)$, when either the reduction method or the explicit method is used. However, when $d \neq 0$, all the critical frequencies of $T(s)$ can be computed with sharp error bounds in many cases, as follows:

Suppose that a $m \times m$ matrix $\hat{M}$ is represented in the computer as $M + \Delta$, $|\Delta_{ij}| \leq \epsilon$. Suppose that ϵ and $\max_{1 \leq i, j \leq m} |M_{ij}|$ are known. Varah^[13] has shown that if every square matrix $M + Q$ with $|q_{ij}| \leq \epsilon$ has no nonlinear elementary divisors,^[14] then using M , ϵ , and $\max_{1 \leq i, j \leq m} |M_{ij}|$, the eigenvalues of M can be computed with (often tight) bounds which take into account the round-off errors.

In the case that $d \neq 0$ we are interested in two matrices $\hat{A}$ and $\hat{A}^*$ where

$$T(s) = \hat{a} \times \frac{\det \left[sI_N - \hat{A} + \frac{1}{\hat{d}} \hat{B}\hat{C} \right]}{\det[sI_N - \hat{A}]} = \hat{a} \times \frac{\det[sI_N - \hat{A}^*]}{\det[sI_N - \hat{A}]} \quad (67)$$

The symbol " $\hat{\cdot}$ " denotes the true value as contrasted to that residing in the computer. Assume that the input data are exact. For the matrix $A = \hat{A}$, $\epsilon_A = 0$ by assumption, and $\max_{1 \leq i, j \leq N} |a_{ij}|$ is readily available. Therefore if

A has no nonlinear elementary divisors we can compute the eigenvalues of A with (often helpful) bounds.

For the matrix $\hat{A}^* = \begin{bmatrix} \hat{a}_{ij}^* \end{bmatrix}$ defined by

$$\hat{a}_{ij}^* = a_{ij} - \frac{1}{\hat{d}} b_i c_j \quad (68)$$

the computed matrix $A^* = \begin{bmatrix} a_{ij}^* \end{bmatrix}$ is given by

$$a_{ij}^* = fl \left(a_{ij} - \frac{1}{\hat{d}} b_i c_j \right) \quad (69)$$

where $fl(\cdot)$ denotes the computer number associated with $(\cdot)$. We assume for simplicity of discussion that the computer employs single-precision arithmetic and single-precision floating-point numbers with a t -bit mantissa.

Clearly $\max_{1 \leq i, j \leq N} |a_{ij}^*|$ is readily available. To obtain $\epsilon_{A^*} \geq \max_{1 \leq i, j \leq N} |a_{ij}^* - a_{ij}^*|$, we follow Wilkinson^[15]

$$a_{ij}^* = \left[a_{ij} - \frac{b_{ij} c_j}{d} (1 + \epsilon_1)(1 + \epsilon_2) \right] (1 + \epsilon_3) \quad (70)$$

in which $|\epsilon_i| \leq 2^{-t}$ for $i = 1, 2, 3$. Thus

$$|a_{ij}^* - a_{ij}^*| = \left| a_{ij} \epsilon_3 - \frac{b_{ij} c_j}{d} [(1 + \epsilon_1)(1 + \epsilon_2)(1 + \epsilon_3) - 1] \right| \quad (71)$$

$$= \left| a_{ij} \epsilon_3 - \frac{b_{ij} c_j}{d} \epsilon_4 \right| \quad (72)$$

where

$$(1 - 2^{-t})^3 - 1 \leq \epsilon_4 \leq (1 + 2^{-t})^3 - 1. \quad (73)$$

Let

$$\epsilon_5 = \max(|(1 + 2^{-t})^3 - 1|, |(1 - 2^{-t})^3 - 1|) \quad (74)$$

Then

$$|a_{ij}^* - a_{ij}^*| \leq \epsilon_3 |a_{ij}| + \epsilon_5 \left| \frac{b_{ij} c_j}{d} \right| \quad (75)$$

We can set

$$\epsilon_{A^*} = \max_{1 \leq i, j \leq N} \left(\epsilon_3 |a_{ij}| + \epsilon_5 \left| \frac{b_{ij} c_j}{d} \right| \right) \quad (76)$$

Therefore, assuming that A^* is "not close" to a matrix possessing nonlinear elementary divisors, we can use (76) and $\max_{1 \leq i, j \leq N} |a_{ij}^*|$ to compute bounds on the computed zeros of $T(s)$.

In Example 1, the driving-point impedance is given by:

$$Z_{in}(s) = \frac{\det[sI_{N-1} - \bar{1}\bar{1}^T A]}{\det[sI_N - A]}.$$

Since no round-off errors were introduced in obtaining A^* in this case, the bounds for the zeros and poles of $Z_{in}(s)$ were computed by setting $\epsilon_A = \epsilon_{A^*} = 0$. In this example each critical frequency was determined to within a radius of 10^{-7} for the input data given.

(b) Frequency Analysis of Networks:

When a network function $T(s)$ is to be evaluated at many frequencies, the traditional method of applying Gaussian elimination to the complex steady-state circuit equations is too time-consuming since the matrix reduction has to be performed for each different frequency. Branin^[16] suggested that it would be more efficient to diagonalize the matrix A first:

$$A = X\Lambda_A X^{-1} \quad (77)$$

$$\Lambda_A = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_N]. \quad (78)$$

Then the network function would be computed at different frequencies:

$$\begin{aligned} T(s) &= C[sI_N - A]^{-1}B + d \\ &= CX[sI_N - \Lambda_A]^{-1}X^{-1}B + d. \end{aligned} \quad (79)$$

Computation efficiency would result from the fact that at each frequency only a diagonal matrix would be inverted.

Branin's suggestion constitutes an improvement. However, the transformation of A to its diagonal, or more generally to its Jordan canonical form is also unnecessarily time-consuming. Furthermore, if some eigenvectors of A are nearly parallel, X will be ill-conditioned. An attractive alternative is to use the techniques of this paper to obtain all the critical frequencies of $T(s)$ first. The computation of the network function at each frequency is again trivial. We note, however, that the eigenvalues are not difficult to find while the eigenvectors may be quite difficult to compute. Furthermore, the inversion of ill-conditioned X is avoided.

Three circuits of increasing complexity are shown in Figs. 2, 3, and 4. The order of the A -matrix for each circuit is, respectively: 6, 8, and 18. Each circuit was analyzed at one hundred frequencies using both the iterative reduction and the explicit matrix formulation methods presented here. For comparison, each circuit was also analyzed at the same frequencies using Gaussian elimination. The computation times, in seconds, are as follows:

Method Circuit	Iterative Reduction	Explicit Formulation	Gauss Elimination
Fig. 2	4.3	3.6	5.7
Fig. 3	5.4	4.3	18.2
Fig. 4	13.3	11.9	81.0

For the circuits considered, the two techniques presented here were approximately equal. Compared to the Gauss elimination approach, both techniques led to a reduction in the computation time⁷: a reduction of seven to one was obtained for the most complex circuit (Fig. 4) analyzed.

However, there are special circuits for which the Gaussian method is faster. These circuits typically have many series (and/or parallel) reactive elements so that the dimension of the A matrix is much greater than that of the steady-state admittance or impedance matrix. For example, in the all-pass lattice^[18] network of Fig. 5, the order of A increases with the order of the lattice while the order of the steady-state immittance matrices are three after impedances of each arm are combined.

APPENDIX A

We wish to prove that, for the linear, time-invariant system

$$\dot{x} = Ax + Bu \quad (A1)$$

$$y = Cx$$

if

$$c_\ell b_\ell \neq 0 \quad \text{for some } \ell \quad (A3)$$

$$c_j b_j = 0 \quad \text{for all } j \neq \ell \quad (A4)$$

then

$$T(s) = c_\ell b_\ell \frac{\det [sI_N - A']_\ell}{\det [sI_N - A']} \quad (A5)$$

where

$$A' = \theta_2 \theta_1^{-1} A \theta_1 \theta_2^{-1} \quad (A6)$$

and θ_1 and θ_2 are elementary transformation matrices such that

$$C\theta_1 = c_{\ell} v_N^{(\ell)} = [0, \dots, 0, c_{\ell}, 0, \dots, 0] \quad (A7)$$

$$\theta_2^B = b_{\ell} \psi_N^{(\ell)} = [0, \dots, 0, b_{\ell}, 0, \dots, 0]^t. \quad (A8)$$

We first note that if (A3) and (A4) hold, then the transformation of variables $\omega = \theta_2 \theta_1^{-1} x$ takes (A1) and (A2) into:

$$\dot{\omega} = A' \omega + b_{\ell} \psi_N^{(\ell)} u \quad (A9)$$

$$y = c_{\ell} v_N^{(\ell)} \omega \quad (A10)$$

since

$$\dot{\omega} = \theta_2 \theta_1^{-1} A \theta_1 \theta_2^{-1} \omega + \theta_2 \theta_1^{-1} B u \quad (A11)$$

$$y = C \theta_1 \theta_2^{-1} \omega \quad (A12)$$

and (as can be easily verified) Equations (A3) and (A4) imply

$$\theta_2 (\theta_1^{-1} B) = \theta_2^B = b_{\ell} \psi_N^{(\ell)} \quad (A13)$$

and

$$(C \theta_1) \theta_2^{-1} = \left(c_{\ell} v_N^{(\ell)} \right) \theta_2^{-1} = c_{\ell} v_N^{(\ell)}. \quad (A14)$$

By applying Theorem 2 to (A9) and (A10) we have

$$T(s) = \frac{Y(s)}{U(s)} = c_{\ell} b_{\ell} \frac{\det \left[I_{N-1} s - \left(I_N - \psi_N^{(\ell)} v_N^{(\ell)} \right) A' \left(I_N - \psi_N^{(\ell)} v_N^{(\ell)} \right) \right]}{\det [I_N - A']} \quad (A15)$$

which leads directly to (A5).

FOOTNOTES

1. In order to simplify the discussion we assume that every eigenvalue of A is a pole of $T(s)$. This assumption is satisfied whenever the system is completely controllable and observable [6].
2. Note that A^* is not unique since the right side of (16) is unchanged if A^* is replaced by any matrix similar to A^* .

3. The notations $V_N^{(\ell)}$, $\bar{v}(\cdot)$, and $(\cdot)_{\bar{v}}$ are defined in Section 3.
4. Equation (41) has also been noted by Brockett^[12] whose derivation is very different.
5. The proof is given in Appendix A.
6. See Section 6(a)(iii).
7. We acknowledge here that in independent work C. Pottle also has considered the problem of determining critical frequencies for circuit analysis. His approach involves considering "the original network connected in the feedback loop of an ideal infinite-gain operational amplifier...." See [17].

REFERENCES

- [1] A. S. Householder, The Theory of Matrices in Numerical Analysis. New York: Blaisdell, 1964, pp. 166-168.
- [2] L. A. Zadeh and C. A. Desoer, Linear System Theory. New York: McGraw-Hill, 1963, pp. 303-305.
- [3] Op. cit., [1] p. 149.
- [4] J. H. Wilkinson, The Algebraic Eigenvalue Problem. New York: Oxford University Press, 1965, pp. 434-435.
- [5] F. J. MacWilliams, "Topological Network Analysis as a Computer Program", IRE Transactions on Circuit Theory, (Correspondence), Vol. CT-5, pp. 228-229, September, 1958.
- [6] R. E. Kalman, "The Mathematical Description of Linear Dynamical Systems," SIAM Journal on Control, Vol. 1, 1963, pp. 152-192.
- [7] Op. cit., [4], chapter 8.
- [8] I. W. Sandberg, "On the Theory of Linear Multi-Loop Feedback Systems," Bell System Technical Journal, Vol. 42, pp. 355-382, March, 1963.
- [9] R. B. Blackman, "Effect of Feedback on Impedance," Bell System Technical Journal, Vol. 22, pp. 268-277, October, 1943.
- [10] S. Lefschetz, Stability of Nonlinear Control Systems. New York: Academic Press, 1965, pp. 132-133.
- [11] D. K. Faddeev and V. N. Faddeeva, Computational Methods of Linear Algebra. San Francisco: Freeman, 1963, p. 113.
- [12] R. W. Brockett, "Poles, Zeros, and Feedback: State Space Interpretation," IEEE Transactions on Automatic Control, Vol. AC-10, pp. 129-135, April, 1965.
- [13] J. M. Varah, "The Computation of Bounds for the Invariant Subspaces of a General Matrix Operator," Ph.D. dissertation, Stanford University, Stanford, 1967.

- [14] F. R. Gantmacher, Matrix Theory. New York: Chelsea, 1960, Vol. 1, p. 152.
- [15] J. H. Wilkinson, Rounding Errors in Algebraic Processes. Englewood Cliffs: Prentice-Hall, 1963, pp. 7-11.
- [16] F. H. Branin, Jr., "Computer Methods of Network Analysis," IEEE Proceedings, Vol. 55, pp. 1787-1801, November, 1967.
- [17] C. Pottle, "On the Determination of Network Zeros by Digital Computer Using the State-Space Approach," (an abstract), Proceedings of the Princeton Conference on Information Sciences and Systems, p. 407, 1968.
- [18] L. Weinberg, Network Analysis and Synthesis. New York: McGraw-Hill, 1962, pp. 630-631.

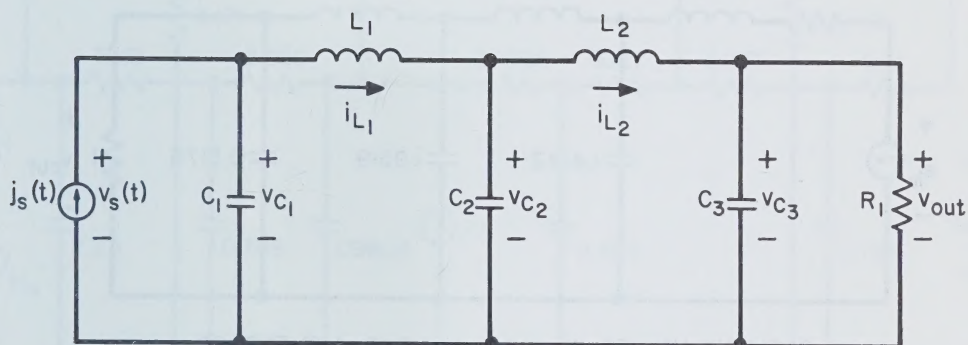
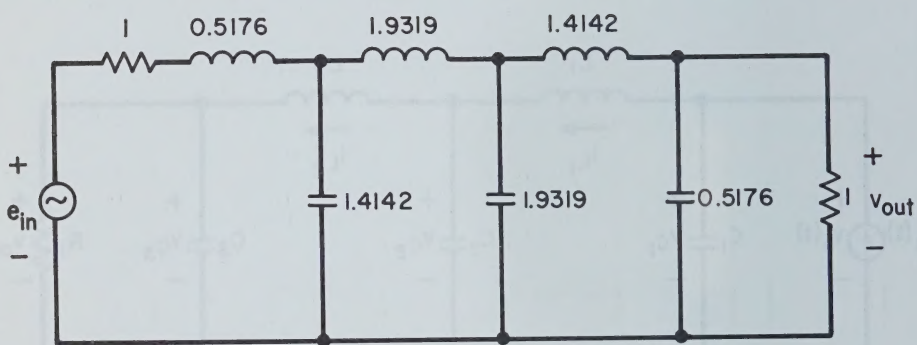
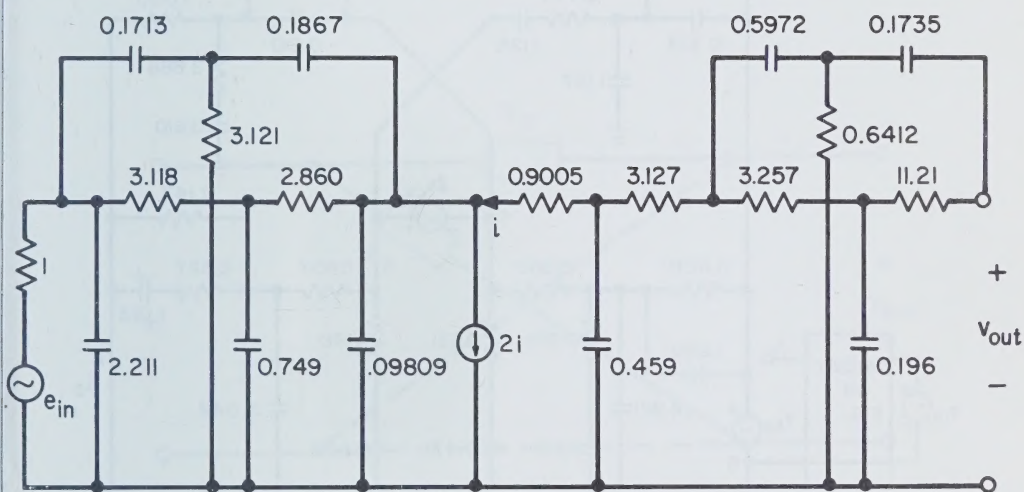


FIG. 1



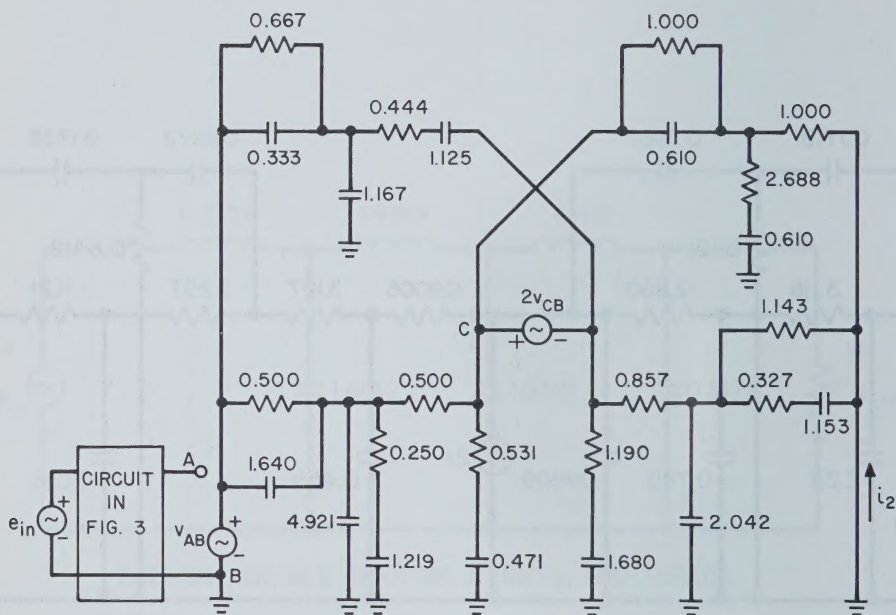
ELEMENT VALUES IN OHMS, HENRYS, AND FARADS

FIG. 2



ELEMENT VALUES IN OHMS AND FARADS

FIG. 3 $T(s) = V_{out}(s)/E_{in}(s)$



ELEMENT VALUES IN OHMS AND FARADS

FIG.4 $T(s) = I_2(s)/E_{in}(s)$

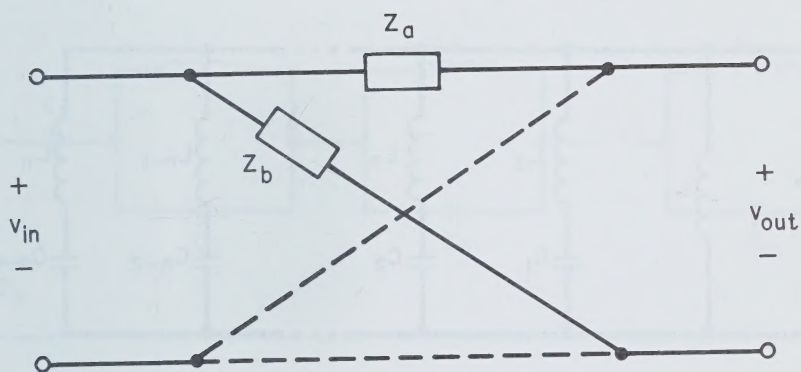


FIG. 5 (a)

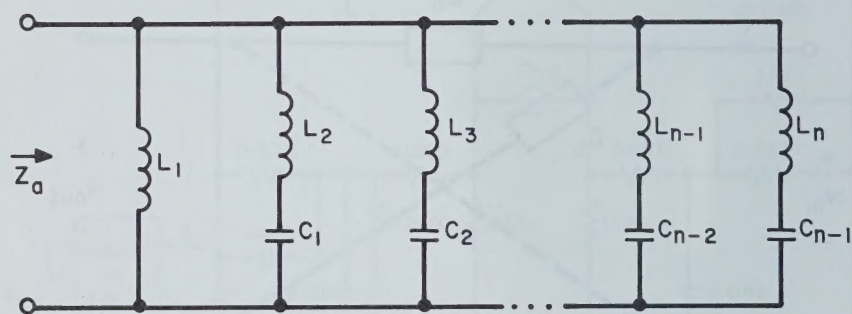


FIG. 5 (b), n IS ODD

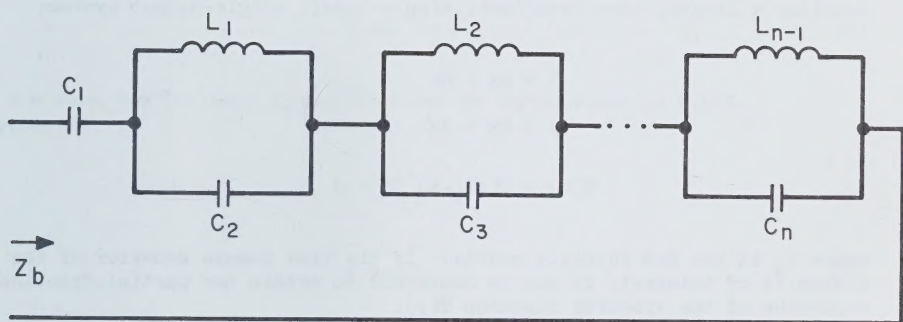


FIG. 5 (c), n IS ODD

A METHOD FOR COMPUTING THE TRANSFER-FUNCTION PARTIAL-FRACTION EXPANSION FROM THE STATE EQUATIONS

H. C. SO

Bell Telephone Laboratories, Incorporated
Murray Hill, New Jersey

ABSTRACT

This paper presents a new method for the partial fraction expansion of the transfer function of a linear, time-invariant, single-input, single-output system. Unlike most existing techniques, neither the transfer function nor $(sI_N - A)^{-1}$ is required to be known or calculated. The method uses the state and output equations directly, assuming that the eigenvalues of A are to be computed with the QR algorithm.

The method is valid regardless of whether multiple poles are present.

1. INTRODUCTION

Consider a linear, time-invariant, single-input, single-output system:

$$\dot{x} = Ax + Bu \quad (1)$$

$$y = Cx + du \quad (2)$$

$$T(s) = C(sI_N - A)^{-1}B + d \quad (3)$$

where I_N is the $N \times N$ identity matrix. If the time domain behavior of the system is of interest, it may be desirable to obtain the partial-fraction expansion of the transfer function $T(s)$:

$$T(s) = d + \sum_{i=1}^g \sum_{\sigma=1}^m \frac{K_i^{(\sigma)}}{(s - \lambda_i)^\sigma} \quad (4)$$

where g is the number of distinct poles of $T(s)$. Although much^[1-4] has been written about the partial-fraction expansion of a transfer function, most techniques assume that $T(s)$, or $(sI_N - A)^{-1}$, is given. In practice, however, the known quantities are the state equations (1) and (2). This paper presents a new method which operates directly on the matrices A , B , C , and d . It is assumed that the eigenvalues of A are found by similarity transformations.

2. PRELIMINARY CONSIDERATION

We begin by rewriting Equation (3) as

$$T(s) = \frac{\det[sI_N - A] \times \det[d + C(sI_N - A)^{-1}B]}{\det[sI_N - A]} \quad (5)$$

Using a well-known determinantal identity,^[5] we have

$$T(s) = \frac{\det \begin{bmatrix} d & -C \\ B & sI_N - A \end{bmatrix}}{\det[sI_N - A]} \quad (6)$$

Suppose now that the eigenvalues of A are found by some method, such as the QR algorithm,^[6] which transforms A into the lower triangular form by means of similarity transformations:

$$T_L A T_L^{-1} = L \quad (7)$$

Let the same row (column) transformations be carried out on B (-C, respectively):

$$\beta = T_L B \quad (8)$$

$$\gamma = -C T_L^{-1} \quad (9)$$

Then

$$T(s) = \frac{\det \left(\begin{bmatrix} 1 & 0 \\ 0 & T_L \end{bmatrix} \begin{bmatrix} d & -C \\ B & sI_N - A \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & T_L^{-1} \end{bmatrix} \right)}{\det[sI_N - A]} \quad (10)$$

$$= \frac{\det \begin{bmatrix} d & \gamma \\ \beta & sI_N - L \end{bmatrix}}{\det[sI_N - A]} \quad (11)$$

Let the diagonal elements of L be denoted by $p_1, p_2, \dots, p_N$; we note that these are the eigenvalues of A. Hence

$$T(s) = (-1)^N \frac{\det \begin{bmatrix} \beta & sI_N - L \\ d & \gamma \end{bmatrix}}{\det[sI_N - P]} = (-1)^N \frac{\det[H]}{\prod_{i=1}^N (s-p_i)} \quad (12)$$

where $P = \text{diag}[p_1 p_2 \dots p_N]$.

The $(N+1) \times (N+1)$ matrix H in the numerator of (12) is a lower Hessenberg matrix^[7] as shown in (13). Note that $h_{ij} = 0$ for $j > i + 1$, and $h_{i \ i+1} = (s-p_i)$.

$$H = [h_{ij}] = \begin{bmatrix} h_{11} & (s-p_1) & & & & \\ h_{21} & h_{22} & (s-p_2) & & & \\ h_{31} & h_{32} & h_{33} & (s-p_3) & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ h_{N1} & h_{N2} & h_{N3} & \dots & h_{NN} & (s-p_N) \\ h_{N+1 \ 1} & h_{N+1 \ 2} & h_{N+1 \ 3} & \dots & h_{N+1 \ N} & h_{N+1 \ N+1} \end{bmatrix} \quad (13)$$

We observe that as long as the eigenvalues of A are computed by a process described by (7), H can be obtained with little additional effort. Applying Hyman's method^[8] for the evaluation of the determinant of a Hessenberg matrix, we define

$$x_1 = 1 \quad (14)$$

$$x_j = \frac{- \sum_{t=1}^{j-1} h_{j-1 \ t} x_t}{s - p_{j-1}}, \quad 2 \leq j \leq N+1 \quad (15)$$

so that

$$T(s) = \sum_{t=1}^{N+1} h_{N+1-t} x_t \quad (16)$$

Equations (14)-(16) form our basis for obtaining the partial-fraction expansion of a transfer function.

3. PARTIAL-FRACTION EXPANSION: GENERAL

Let A have g distinct eigenvalues. Let these distinct eigenvalues be labeled $\lambda_1, \lambda_2, \dots, \lambda_g$, in the order that they appear on the superdiagonal of H , from h_{12}^g to h_{N+1}^g . For each $x_t(s)$, let $m_{t,i}$ be the order of the pole at $(s-\lambda_i)$. Define the pole set S_t of $x_t(s)$ as a subset of $\{\lambda_1, \lambda_2, \dots, \lambda_g\}$ such that $\lambda_i \in S_t$ whenever there exists a μ such that

$$p_\mu = \lambda_i, \quad 1 \leq \mu \leq t-1. \quad (17)$$

In view of these notations, the partial-fraction expansion of any x_t may be written as

$$x_t(s) = k_t^{(0)} + \sum_i \sum_{\substack{\sigma=1 \\ \lambda_i \in S_t}}^{m_{t,i}} \frac{k_{t,i}^{(\sigma)}}{(s-\lambda_i)^\sigma} \quad 1 \leq t \leq N+1 \quad (18)$$

where $k_t^{(0)}$ and all $k_{t,i}^{(\sigma)}$'s are constants. Note that $k_{t,i}^{(\sigma)} = 0$ for $t \leq i$. Now from (15),

$$x_j(s) = \frac{-1}{s-p_{j-1}} \sum_{t=1}^{j-1} h_{j-1-t} \left[k_t^{(0)} + \sum_i \sum_{\substack{\sigma=1 \\ \lambda_i \in S_t}}^{m_{t,i}} \frac{k_{t,i}^{(\sigma)}}{(s-\lambda_i)^\sigma} \right] \quad (19)$$

$$= \frac{-1}{s-p_{j-1}} \left[h_{j-1-1} + \sum_i \sum_{\substack{j,i \\ \lambda_i \in Q_j}}^{M_{j,i}} \frac{\sum_{t=2}^{j-1} h_{j-1-t} k_{t,i}^{(\sigma)}}{(s-\lambda_i)^\sigma} \right] \quad (20)$$

with $M_{j,i} = \max_{2 \leq t \leq j-1} (m_{t,i})$, and $Q_j = S_2 \cup S_3 \cup S_4 \dots \cup S_{j-1}$. Let

$$\theta_{j,i}^{(\sigma)} = \sum_{t=2}^{j-1} h_{j-1,t} k_{t,i}^{(\sigma)} \quad (21)$$

$$\therefore x_j(s) = \frac{-h_{j-1,1}}{s-p_{j-1}} - \sum_{\substack{i \\ \lambda_i \subset Q_j}} \sum_{\sigma=1}^{M_{j,i}} \frac{\theta_{j,i}^{(\sigma)}}{(s-\lambda_i)^{\sigma}(s-p_{j-1})} \quad (22)$$

If $p_{j-1} \neq \lambda_i$ for all $\lambda_i \subset Q_j$, then

$$x_j(s) = \frac{-h_{j-1,1}}{s-p_{j-1}} - \sum_{\substack{i \\ \lambda_i \subset Q_j}} \sum_{\sigma=1}^{M_{j,i}} \left[\frac{\theta_{j,i}^{(\sigma)} / (p_{j-1} - \lambda_i)^{\sigma}}{s-p_{j-1}} + \sum_{\alpha=1}^{\sigma} \frac{\sigma_{j,i}^{w(\alpha)}}{(s-\lambda_i)^{\alpha}} \right] \quad (23)$$

where

$$\sigma_{j,i}^{w(\alpha)} = \frac{1}{(\sigma-\alpha)!} \frac{d^{(\sigma-\alpha)}}{ds^{(\sigma-\alpha)}} \left[\theta_{j,i}^{(\sigma)} (s-p_{j-1})^{-1} \right] \Big|_{s=\lambda_i} \quad \text{for } 1 \leq \alpha \leq \sigma \quad (24)$$

$$\sigma_{j,i}^{w(\alpha)} = 0 \quad \text{for } \alpha > \sigma \quad (25)$$

Hence

$$\sigma_{j,i}^{w(\sigma)} = \frac{\theta_{j,i}^{(\sigma)}}{\lambda_i - p_{j-1}}$$

$$\sigma_{j,i}^{w(\sigma-1)} = \frac{(-1)^{\theta_{j,i}^{(\sigma)}}}{(\lambda_i - p_{j-1})^2}$$

⋮

$$\sigma_{j,i}^{w(\alpha)} = \frac{(-1)^{\sigma_{j,i}^{w(\alpha)+1}}}{\lambda_i - p_{j-1}} \quad \text{for } 1 \leq \alpha < \sigma \quad (26)$$

Therefore if $p_{j-1} \neq \lambda_i$ for all $\lambda_i \in Q_j$,

$$x_j(s) = \frac{-1}{s - p_{j-1}} \left[h_{j-1} + \sum_{\substack{i \\ \lambda_i \in Q_j}} \sum_{\sigma=1}^{M_{j,i}} \frac{\theta_{j,i}^{(\sigma)}}{(p_{j-1} - \lambda_i)^\sigma} \right] - \sum_{\substack{i \\ \lambda_i \in Q_j}} \sum_{\sigma=1}^{M_{j,i}} \sum_{\alpha=1}^{\sigma} \frac{\sigma_{j,i}^{w(\alpha)}}{(s - \lambda_i)^\alpha} \quad (27)$$

If $p_{j-1} = \lambda_w$ for some $\lambda_w \in Q_j$,

$$x_j(s) = \frac{-1}{s - p_{j-1}} \left[h_{j-1} + \sum_{\substack{i \neq w \\ \lambda_i \in Q_j}} \sum_{\sigma=1}^{M_{j,i}} \frac{\theta_{j,i}^{(\sigma)}}{(p_{j-1} - \lambda_i)^\sigma} \right] - \sum_{\substack{i \neq w \\ \lambda_i \in Q_j}} \sum_{\sigma=1}^{M_{j,i}} \sum_{\alpha=1}^{\sigma} \frac{\sigma_{j,i}^{w(\alpha)}}{(s - \lambda_i)^\alpha} \\ - \sum_{\sigma=1}^{M_{j,w}} \frac{\theta_{j,w}^{(\sigma)}}{(s - \lambda_w)^{\sigma+1}} \quad (28)$$

Comparing (27) and (28) with the expression for $x_j(s)$,

$$x_j(s) = \sum_{\lambda_i \in S_j} \sum_{\sigma=1}^{m_{j,i}} \frac{k_{j,i}^{(\sigma)}}{(s - \lambda_i)^\sigma} \quad (29)$$

it is seen that if $p_{j-1} = \lambda_w$, then

$$k_{j,i}^{(\alpha)} = - \sum_{\sigma=\alpha}^{M_{j,i}} \sigma_{j,i}^{w(\alpha)} \quad \lambda_i \subset Q_j, \quad i \neq w, \quad 1 \leq \alpha \leq M_{j,i} \quad (30)$$

$$k_{j,w}^{(1)} = - \left[h_{j-1,1} - \sum_{\substack{i \neq w \\ \lambda_i \subset Q_j}} \sum_{\sigma=1}^{M_{j,i}} \sigma_{j,i}^{w(1)} \right] \quad (31)$$

$$k_{j,w}^{(\alpha)} = -\theta_{j,w}^{(\alpha-1)} \quad \text{for } 1 < \alpha \leq M_{j,w} + 1 \quad \text{if } \lambda_w \subset Q_j \quad (32)$$

$$k_{j,w}^{(\alpha)} = 0 \quad \text{for } 1 < \alpha \quad \text{if } \lambda_w \not\subset Q_j \quad (33)$$

Equations (30)-(33) may be rewritten as:

$$k_{j,i}^{(M_{j,i})} = \frac{-1}{\lambda_i - \lambda_w} \sum_{t=2}^{j-1} h_{j-1,t} k_{t,i}^{(M_{j,i})}, \quad \lambda_i \subset Q_j, \quad i \neq w. \quad (34)$$

$$k_{j,i}^{(\alpha)} = \frac{-1}{\lambda_i - \lambda_w} \left(k_{j,i}^{(\alpha+1)} + \sum_{t=2}^{j-1} h_{j-1,t} k_{t,i}^{(\alpha)} \right), \quad \lambda_i \subset Q_j, \quad i \neq w, \quad 1 \leq \alpha < M_{j,i}. \quad (35)$$

$$k_{j,w}^{(1)} = - \left(h_{j-1,1} + \sum_{\substack{i \neq w \\ \lambda_i \subset Q_j}} k_{j,i}^{(1)} \right) \quad (36)$$

$$k_{j,w}^{(\alpha)} = - \sum_{t=2}^{j-1} h_{j-1,t} k_{t,w}^{(\alpha-1)}, \quad 1 < \alpha \quad (37)$$

Equations (34)-(37) are the recursion formulae which give the partial-fraction expansion of $x_j(s)$ in terms of the expansions of $x_i(s)$, $1 \leq i < j$. At the end, the expansion of $T(s)$ is obtained by summing all corresponding terms:

$$T(s) = \sum_{t=1}^{N+1} h_{N+1-t} x_t \quad (38)$$

$$= h_{N+1-1} + \sum_{i=1}^{g_1} \sum_{\sigma=1}^{M_1} h_{N+1-t} \frac{k_{t,i}^{(\sigma)}}{(s-\lambda_1)^\sigma} \quad (39)$$

where $M_1 = \max_{2 \leq t \leq N+1} (m_{t,i})$.

Example 1: Consider the partial-fraction expansion of the function

$$T(s) = \frac{1}{(s-1)^2(s-2)^2(s-3)} \times \det \begin{bmatrix} 1 & s-1 & & & & \\ & 1 & 2 & s-2 & & \\ & 1 & 2 & 2 & s-2 & \\ & 1 & 0 & 0 & 1 & s-1 \\ & 1 & 1 & 1 & 1 & 1 & s-3 \\ & 1 & 2 & -1 & -2 & 3 & 4 \end{bmatrix}$$

Let $\lambda_1 = 1$, $\lambda_2 = 2$, and $\lambda_3 = 3$. We have:

$$x_1 = 1$$

$$x_2: k_{2,1}^{(1)} = -h_{11} = -1$$

$$x_3: k_{3,1}^{(1)} = -\frac{1}{\lambda_1 - \lambda_2} \left(h_{21} k_{1,1}^{(1)} + h_{22} k_{2,1}^{(1)} \right) = -2$$

$$k_{3,2}^{(1)} = - \left(h_{21} + k_{3,1}^{(1)} \right) = 1$$

$$x_4: k_{4,1}^{(1)} = \frac{-1}{\lambda_1 - \lambda_2} \left(\sum_{t=2}^3 h_{3t} k_{t,1}^{(1)} \right) = -6$$

$$k_{4,2}^{(2)} = - \sum_{t=2}^3 h_{3t} k_{t,2}^{(1)} = -2$$

$$k_{4,2}^{(1)} = - \left(h_{31} + k_{4,1}^{(1)} \right) = 5$$

$$x_5: \quad k_{5,2}^{(2)} = \frac{-1}{\lambda_2 - \lambda_1} \left(\sum_{t=2}^4 h_{4t} k_{t,2}^{(2)} \right) = 2$$

$$k_{5,2}^{(1)} = \frac{-1}{\lambda_2 - \lambda_1} \left(k_{5,2}^{(2)} + \sum_{t=2}^4 h_{4t} k_{t,2}^{(1)} \right) = -7$$

$$k_{5,1}^{(1)} = - \left(h_{41} + k_{5,2}^{(1)} \right) = 6$$

$$k_{5,1}^{(2)} = - \sum_{t=2}^4 h_{4t} k_{t,1}^{(1)} = 6$$

$$x_6: \quad k_{6,1}^{(2)} = \frac{-1}{\lambda_1 - \lambda_3} \left(\sum_{t=2}^5 h_{5t} k_{t,1}^{(2)} \right) = 3$$

$$k_{6,1}^{(1)} = \frac{-1}{\lambda_1 - \lambda_3} \left(k_{6,1}^{(2)} + \sum_{t=2}^5 h_{5t} k_{t,1}^{(1)} \right) = 0$$

$$k_{6,2}^{(2)} = \frac{-1}{\lambda_1 - \lambda_3} \left(\sum_{t=2}^5 h_{5t} k_{t,2}^{(2)} \right) = 0$$

$$k_{6,2}^{(1)} = \frac{-1}{\lambda_2 - \lambda_3} \left(\sum_{t=2}^5 h_{5t} k_{t,2}^{(1)} \right) = -1$$

$$k_{6,3}^{(1)} = - \left(h_{51} + k_{6,1}^{(1)} + k_{6,2}^{(1)} \right) = 0$$

$$T(s) = h_{61} + \frac{K_1^{(1)}}{s-1} + \frac{K_1^{(2)}}{(s-1)^2} + \frac{K_2^{(1)}}{(s-2)} + \frac{K_2^{(2)}}{(s-2)^2}$$

$$K_1^{(1)} = \sum_{t=2}^6 h_{6t} k_{t,1}^{(1)} = 30$$

$$K_1^{(2)} = \sum_{t=2}^6 h_{6t} k_{t,1}^{(2)} = 30$$

$$K_2^{(1)} = \sum_{t=2}^6 h_{6t} k_{t,2}^{(1)} = -36$$

$$K_2^{(2)} = \sum_{t=2}^6 h_{6t} k_{t,2}^{(2)} = 10$$

4. PARTIAL-FRACTION EXPANSION: SPECIAL CASE OF ALL SIMPLE POLES

From the definition of $x_j(s)$ in (15), it is evident that, when all the eigenvalues of A are simple,

$$k_{j,i}^{(1)} = \frac{-1}{\lambda_i - \lambda_{j-1}} \sum_{t=2}^{j-1} h_{j-1,t} k_{t,i}^{(1)} \quad 1 \leq i < j-1 \quad (40)$$

$$\begin{aligned} k_{j,j-1}^{(1)} &= - \sum_{t=1}^{j-1} h_{j-1,t} x_t(\lambda_{j-1}) \\ &= - \left(h_{j-1,1} + \sum_{i=1}^{j-2} k_{j,i}^{(1)} \right) \end{aligned} \quad (41)$$

The last two equations are, necessarily, special forms of (34)-(37). Since $T(s)$ has only simple poles, however, the residue at each pole may be interpreted very simply. Thus the residue of $T(s)$ at any simple pole p_j is given by:

$$K_{p_j} = (-1)^N \frac{\det[H]}{\prod_{i \neq j} (s - p_i)} \bigg|_{s=p_j} \quad (42)$$

It is recognized that at $s = p_j$ the matrix H contains a zero in the superdiagonal at the $j, j+1$ position. Hence

$$\det[H]_{s=p_j} = \det[E]_{s=p_j} \times \det[F]_{s=p_j} \quad (43)$$

where E (or F) is the submatrix consisting of the first j rows and columns (or last N+1-j rows and columns, respectively,) of H. Both E and F are lower Hessenberg matrices and their determinants may be evaluated at $s = p_j$ readily.

5. CONCLUSION

If Equations (40) and (41) are used to find the complete partial-fraction expansion of a transfer function T(s) having only simple poles, the number of multiplications required is given by:

$$N_M = \frac{N^2 - N}{2} + 1 + \sum_{i=2}^N \frac{i(i+1)}{2} \quad (44)$$

The number of additions required is given by:

$$N_S = \frac{5}{4} N(N-1) + \frac{1}{2} \sum_{i=1}^{N-1} i^2 \quad (45)$$

These requirements are quite low. Furthermore, a comparison of Equations (34)-(37) with (40) and (41) shows that the required operations for an N-order system with multiple poles are virtually the same as those given in (44) and (45).

In view of the fact that the residues are obtained by applying Hyman's method for the evaluation of the determinant of a Hessenberg matrix, it is possible to adopt results^[9] showing the effects of round-off errors on Hyman's method to the present case as well. Such analysis will only provide a partial answer to the question of accuracy of the computed residues since the effects of imperfect similarity transformations on the system matrices will not be accounted for.

REFERENCES

1. K. R. Rao and N. Ahmed, "Recursive Techniques for Obtaining the Partial Fraction Expansion of a Rational Function," Proceedings of the First Asilomar Conference on Circuits and Systems, pp. 522-529, 1967.
2. C. Pottle, "On the Partial Fraction Expansion of a Rational Function with Multiple Poles by Digital Computer," IEEE Transactions on Circuit Theory (Correspondence) vol. CT-11, pp. 161-162, March 1964.
3. H. M. Power, "Useful Relations for Partial Expansion of Proper Rational Functions and Transition Matrices," IEEE Transactions on Education (Short Papers), vol. E-10, pp. 179-180, September 1967.

4. N. Ahmed and K. R. Rao, "Partial Fraction Expansion of Rational Functions with One High-Order Pole," IEEE Transactions on Automatic Control (Correspondence), vol. AC-13, p. 133, February 1968.
5. S. Lefschetz, Stability of Nonlinear Control Systems. New York: Academic Press, 1965, pp. 132-133.
6. J. H. Wilkinson, The Algebraic Eigenvalue Problem. New York: Oxford University Press, 1965, Chapter 8.
7. Op. cit., [6], pp. 218-219.
8. Op. cit., [6], pp. 426-431.
9. J. H. Wilkinson, Rounding Errors in Algebraic Processes. Englewood Cliffs: Prentice-Hall, 1963, pp. 147-150.

THOMAS G. MARSHALL, JR.
Rutgers - The State University
New Brunswick, New Jersey

ABSTRACT

A vector space algorithm is presented which is analogous to the Euclidean algorithm for finding the greatest common divisor (G.C.D.) of a pair of polynomials. The companion matrix of the G.C.D. of the characteristic polynomials of a pair of matrices is obtained. Special cases of the algorithm provide methods for finding the characteristic polynomial of a matrix and for reducing a matrix to tridiagonal form. Analogies with polynomial division are observed which give new, simplifying viewpoints of these procedures. The relationship to the Krylov and Hessenberg procedures is discussed.

I. INTRODUCTION

In the algorithm to be described, a given pair of matrices or vectors determines a block-tridiagonal matrix. If the (unknown) characteristic polynomials of the pair of matrices have a common factor, or if the pair of vectors satisfies a related condition, the block-tridiagonal matrix contains a companion matrix which displays the coefficients of this factor.

Two particular cases of this algorithm are of special interest. The first is matrix tridiagonalization which leads to previously discussed applications in stability testing, determining eigensystems, and ladder network synthesis [1,2]. An analogy with polynomial division is observed here which gives a new viewpoint of the tridiagonalization procedure.

The second particular case of the algorithm yields the characteristic polynomial of a matrix. In this case both matrices of the pair (or both vectors) are chosen to be equal so that the G.C.D. is the unknown characteristic polynomial. The algorithm yields the companion matrix of the G.C.D. which displays the desired coefficients of the characteristic polynomial.

This procedure of obtaining the characteristic polynomial is completely analogous to polynomial division and is therefore simple in concept and execution. The procedure is related to the Krylov and Hessenberg procedures for finding the characteristic polynomial.

II. GREATEST COMMON DIVISOR ALGORITHM

In this section it will first be shown that the execution of the Euclidean algorithm upon a pair of monic polynomials, both of degree n , uniquely determines a pair of $n \times n$ block tridiagonal matrices. Modifications of the Euclidean algorithm are then discussed which permit vectors to be employed instead of polynomials, thus giving the vector space algorithm.

A. The Pair of Matrices

The two matrices are denoted $\underline{T}$ and $\underline{T}^T$. $\underline{T}$ has the block-tridiagonal structure,

$$\underline{T} = \begin{bmatrix} \underline{D}_a & \underline{U}_a & & & \\ & \underline{L}_b & \underline{D}_b & \underline{U}_b & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \underline{U}_l \\ & & & & \underline{L}_m & \underline{D}_m \end{bmatrix}, \quad (2.1)$$

where the only nonzero entries are the indicated submatrices. The $\underline{D}_i$ are square companion matrices, not necessarily of the same size, and the $\underline{L}_i$ and $\underline{U}_i$ are rectangular matrices each of which conforms to the two matrices $\underline{D}_i$ that it borders in row and column size. The subscript which identifies each submatrix is given the value of the row number of $\underline{T}$ preceding the first row of the submatrix. Thus,

$$a = 0 < b < c < \dots < m < n. \quad (2.2)$$

The following example illustrates how $\underline{T}$ might appear for $n = 6$:

$$\underline{T} = \begin{bmatrix} -\delta_{11} & -\delta_{12} & \cdot & & & -x_{15} \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ & & 1 & \cdot & -\delta_{33} & -\delta_{34} & -\delta_{35} & -x_{36} \\ & & & \cdot & 1 & & & \cdot \\ & & & \cdot & & & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & \cdot & & & 1 & -\delta_{66} \end{bmatrix} \quad (2.3)$$

It is seen here, as is true in general, that the submatrices $\underline{L}_i$ and $\underline{U}_i$ are zero except for the entry in the upper, right-hand corner. This entry in $\underline{L}_i$ is always 1 so that $\underline{T}$ has the unit-upper-Hessenberg structure. The notation of δ_{ij} and χ_{ij} for the negatives of those nonzero entries in the upper triangle is convenient and corresponds to the notation in the references for the special case discussed there in which $\underline{T}$ is tridiagonal, i.e. all submatrices are 1×1 .

The matrix $\underline{T}^-$ differs from $\underline{T}$ in at most one entry in $\underline{D}_a$, and the corresponding submatrix of $\underline{T}^-$ is denoted $\underline{D}_a^-$. When $\underline{T}^-$ differs from $\underline{T}$, $\underline{D}_a^-$ is obtained from $\underline{D}_a$ by changing the sign in front of the upper right-hand entry of $\underline{D}_a$, $-\delta_{lb}$, so that the corresponding entry in $\underline{D}_a^-$ is $+\delta_{lb}$. The only circumstance in which $\underline{D}_a$ and $\underline{D}_a^-$ are the same is in the case (to be discussed below) in which $\underline{D}_a = \underline{T}$.

Thus, in this example,

$$\underline{D}_a^- = \underline{D}_0^- = \begin{bmatrix} -\delta_{11} + \delta_{12} \\ 1 \end{bmatrix}. \quad (2.4)$$

B. The Euclidean Algorithm

It will now be shown that $\underline{T}$ and $\underline{T}^-$ are intimately related to the sequence of polynomials that would be obtained if the Euclidean algorithm were executed upon their characteristic polynomials. Let $\underline{T}_i$ and $\underline{T}_i^-$ be submatrices of $\underline{T}$ and $\underline{T}^-$ obtained by deleting the first i rows and columns of each, and let t_i and t_i^- be the corresponding characteristic polynomials in the symbol s , i.e.

$$t_i = \left| s\underline{I}_i - \underline{T}_i \right| \quad (2.5)$$

and

$$t_i^- = \left| s\underline{I}_i - \underline{T}_i^- \right|. \quad (2.6)$$

The degree of t_i or t_i^- is $n - i$. Also let d_i be the characteristic polynomials of $\underline{D}_i$, for example

$$d_a = s^b + \delta_{11}s^{b-1} + \delta_{12}s^{b-2} + \dots + \delta_{lb}. \quad (2.7)$$

The following sequence can be obtained from $\underline{T}$ and $\underline{T}^-$:

$$\begin{aligned}
 t_a^- &= t_a - 2\delta_{lb} t_b \\
 t_a &= d_a t_b + x_{a+1,c} t_c \\
 t_b &= d_b t_c + x_{b+1,d} t_d \\
 &\cdot \\
 &\cdot \\
 t_\ell &= d_\ell t_m + x_{\ell+1,n} t_n \\
 t_m &= d_m t_n,
 \end{aligned} \tag{2.8}$$

where $t_n = 1$. Each equation except the first is obtained by expanding the determinant $t_i = |\underline{I}_i - \underline{T}_i|$ along its top row, which is row $i+1$ in $|\underline{sI} - \underline{T}|$, for $i = a, b, \dots, m$. The first equation is obtained by expanding t_a^- along its top row and comparing with t_a . In (2.3), $a = 0$, $b = \ell = 2$, and $c = m = 5$.

In the special case where $\underline{T} = \underline{T}^-$, the following sequence is obtained:

$$\begin{aligned}
 t_a^- &= t_a \\
 t_a &= d_a t_n.
 \end{aligned} \tag{2.9}$$

This special case is of interest in the problem of finding the characteristic polynomial of a matrix. This is the case referred to in Section IA in which $\underline{D}_a = \underline{T}$.

These sequences have the appearance of the sequence obtained by executing the Euclidean algorithm upon t_0^- and t_0 . Each step of the Euclidean algorithm is polynomial division, say t_i/t_j , which gives an equation of the form

$$t_i = qt_j + r \tag{2.10}$$

The remainder is either zero or else is of degree less than t_j . It is seen that the G.C.D. of t_i and t_j is a factor of r if $r \neq 0$. The equations in (2.8) and (2.9) are of the form (2.10), so the pair $\underline{T}$ and $\underline{T}^-$ determine a Euclidean sequence of polynomials.

The converse of this conclusion is desired, namely that any pair of monic polynomials, say a_0 and b_0 , both of degree n , uniquely determines a pair of matrices, $\underline{T}$ and $\underline{T}^-$. It is easy to see that this is the case: specify

$$t_a = a_0 \quad (2.11)$$

and

$$t_a^- = b_0, \quad (2.12)$$

and perform the Euclidean algorithm upon t_a and t_a^- adopting the notation of either (2.8) or (2.9), depending upon whether t_a and t_a^- are distinct. It follows from the uniqueness of this Euclidean sequence that the entries of $\underline{T}$ and $\underline{T}^-$ are uniquely determined.

Two modifications in the Euclidean algorithm, as it is normally stated, have been introduced here, for convenience. Firstly, the algorithm is stated in terms of monic polynomials. Secondly, when t_a^- and t_a have a nonconstant G.C.D., the scalar $\chi_{\ell+1,n} = 0$ in (2.8). The Euclidean algorithm would normally terminate with this step since a zero remainder has appeared. However, the convention of always including the last equation in (2.8) will be adhered to here, whether a nonconstant G.C.D. is present or not. The sequence (2.9) is a case in point.

C. A Vector Space Analog of the Euclidean Algorithm

The vector space algorithm will be described by indicating how the polynomials in the Euclidean algorithm can be replaced by vectors, so as to give the desired applications, and then showing that the same steps as in the Euclidean algorithm can be performed employing the vectors.

The polynomials in the Euclidean algorithm can be replaced by vectors in a way that was shown in the references to have quite general application. Vectors corresponding to t_i and t_i^- are defined as follows:

$$\underline{y}_i = \underline{v}t_i(\underline{M}) \quad (2.13)$$

and

$$\underline{y}_i^- = \underline{v}t_i^-(\underline{M}). \quad (2.14)$$

The initiating vectors are thus specified by a_0 and b_0 according to

$$\underline{y}_0 = \underline{v}a_0(\underline{M}) \quad (2.15)$$

and

$$\underline{y}_0^- = \underline{v}b_0(\underline{M}). \quad (2.16)$$

$\underline{M}$ and $\underline{y}$ are chosen for convenience subject to the restrictions that $\underline{M}$ be a cyclic (nonderogatory) $n \times n$ matrix and that $\underline{y}$ be a cyclic vector for $\underline{M}$. A more explicit statement of the cyclic property, which is used below, is that $\underline{M}$ be similar to a companion matrix, $\underline{C}$,

$$\underline{XMX}^{-1} = \underline{C}, \quad (2.17)$$

where row i of $\underline{X}$ is

$$\underline{x}_i = \underline{vM}^{n-i}. \quad (2.18)$$

The algorithm can be given an explicit matrix interpretation because the following relations hold and can be verified, as in the references:

$$\underline{YMY}^{-1} = \underline{T} + \underline{e}_1^t \underline{y}_0 \underline{Y}^{-1} \quad (2.19)$$

and

$$\underline{YMY}^{-1} = \underline{T}^- + \underline{e}_1^t \underline{y}_0^- \underline{Y}^{-1}, \quad (2.20)$$

where $\underline{e}_1^t$ is the first column of $\underline{I}$, and $\underline{Y}$ is the matrix with row i equal to $\underline{y}_i$ in (2.13). In the special case where either $\underline{y}_0$ or $\underline{y}_0^-$ is zero, the algorithm reduces $\underline{M}$ to the block-tridiagonal form $\underline{T}$ or $\underline{T}^-$, respectively, by a similarity transformation.

The vectors $\underline{y}_0$ and $\underline{y}_0^-$ which initiate the algorithm can be determined by a pair of $n \times n$ matrices $\underline{A}$ and $\underline{B}$ by specifying a_0 and b_0 as follows:

$$a_0 = |s\underline{I} - \underline{A}| \quad (2.21)$$

and

$$b_0 = |s\underline{I} - \underline{B}|. \quad (2.22)$$

With this interpretation of a_0 and b_0 it is seen that the G.C.D. is indeed that of the characteristic polynomials of $\underline{A}$ and $\underline{B}$. If $\underline{A}$ and $\underline{B}$ are cyclic, (2.11) and (2.12) imply that

$$\underline{A} \sim \underline{T} \quad (2.23)$$

and

$$\underline{B} \sim \underline{T}^-, \quad (2.24)$$

and then the algorithm reduces $\underline{A}$ and $\underline{B}$ to the block-tridiagonal form by similarity transformation.

The vectors $\underline{y}_0$ and $\underline{y}_0^-$ will be assumed known. This does not require that a_0 and b_0 be known. For example, if $\underline{M} = \underline{A}$ the Cayley-Hamilton theorem gives $\underline{y}_0 = 0$, and a_0 need not be known. Sometimes, either $\underline{y}_0$ or $\underline{y}_0^-$ can be arbitrarily specified. In addition, when a_0 and b_0 are given by (2.21) and (2.22) and $\underline{A}$ and $\underline{B}$ are unit-upper-Hessenberg, an algorithm has been given for finding $\underline{y}_0$ and $\underline{y}_0^-$ directly without determining a_0 and b_0 [2].

There are three essential operations to be performed in determining $\underline{T}$ and $\underline{T}^-$ from $\underline{y}_0$ and $\underline{y}_0^-$ instead of from t_0 and t_0^- . Firstly, vectors must be added and subtracted, and this can be done in the familiar way. Secondly, multiplication of a polynomial by s is replaced by post multiplication of the corresponding vector by $\underline{M}$. Lastly, the lead coefficients of certain polynomials must be determined from the corresponding vectors. Namely, the entries $x_{i+1,k}$ which are the lead coefficients of the remainders, r , must be determined from the corresponding vectors $\underline{vr}(\underline{M})$ which appear; and, similarly, the entries δ_{ij} , which are the lead coefficients of the intermediate remainders formed in obtaining the quotients d_i in (2.8) by normal polynomial "long division", must be determined.

Denoting an intermediate remainder of degree $n-j$ as p_j , the corresponding vector is

$$\underline{z}_j \equiv \underline{vp}_j(\underline{M}). \quad (2.25)$$

The lead coefficient is δ_{ij} , and it is given by

$$\delta_{ij} = \underline{z}_j \underline{M}^{j-1} \underline{h}_1^t, \quad (2.26)$$

where $\underline{h}_1^t$ is the first column of $\underline{X}^{-1}$ in (2.17). This relation follows by observing from (2.18) that the vectors $\underline{x}_j$ correspond to polynomials s^{n-j} , and, since

$$\underline{x}_{j-1}^t = 0 \quad (2.27)$$

for $j > 1$ and

$$\underline{x}_{1-1}^t = 1, \quad (2.28)$$

that $\underline{h}_1^t$ selects the coefficient of s^{n-1} in $p_j s^{j-1}$ from the vector $\underline{z}_j \underline{M}^{j-1}$ in (2.26) [1].

These steps are all illustrated in the next section and also appear in the algorithms of the references which pertain to the special case in which $\underline{T}$ is tridiagonal.

III. AN ALGORITHM FOR DETERMINING THE CHARACTERISTIC POLYNOMIAL

A. The Algorithm

In this special case of the previously described algorithm it is assumed that $\underline{B} = \underline{A}$ so that the G.C.D. is a_0 , the unknown characteristic polynomial of $\underline{A}$. The algorithm yields the companion matrix $\underline{D}_a = \underline{T}$, which displays the coefficients of a_0 . If $\underline{A}$ is cyclic, (2.23) also holds.

It is assumed that $\underline{y}_0$ is known, and, in particular, if $\underline{M}$ is chosen to be equal to $\underline{A}$ or to have the same characteristic polynomial, $\underline{y}_0 = 0$. It is also assumed that the cyclic vector, $\underline{v} = \underline{y}_n$ is known. An arbitrary choice can be made for $\underline{v}$, since in the unusual case in which the vector chosen is not cyclic, this will become apparent when $\underline{h}_{-1}^t$ is being determined, allowing another choice to be made.

The vector sequence corresponds to (2.9). Since $a = 0$, it can be written

$$\begin{aligned}\underline{y}_0^- &= \underline{y}_0 \\ \underline{y}_0 &= \underline{y}_n d_0(\underline{M}).\end{aligned}\tag{3.1}$$

The second equation is to be solved for the coefficients δ_{1j} of d_0 . This can be done in a completely analogous manner to the division of t_0 by $t_n = 1$, and the scheme can, therefore, be conveniently depicted as follows:

$$\begin{array}{r} \delta_{11} \quad \delta_{12} \quad . \quad . \quad \delta_{1n} \\ \underline{y}_n \left| \begin{array}{l} \underline{y}_0 \\ \underline{y}_n M^n \end{array} \right. \\ \hline \delta_{11} \underline{y}_n M^{n-1} \quad \underline{z}_1 \\ \hline \delta_{12} \underline{y}_n M^{n-2} \quad \underline{z}_2 \\ \hline \quad \quad \quad \quad \quad \underline{z}_n \\ \delta_{1n} \underline{y}_n M^0 \\ \hline 0 \end{array}\tag{3.2}$$

For purposes of comparison with polynomial division, the vectors are shown indented, even though, in general, all the vectors will have n nonzero entries.

Since $\underline{T} = \underline{D}_0$ is a companion matrix, (2.5) shows that $t_i = s^{n-i}$ for $i = 1, \dots, n$. The corresponding vectors $\underline{y}^i = \underline{y}_n \underline{M}^{n-i}$. These vectors are required in the algorithm as seen in (3.2). From (2.18) it is seen that $\underline{y}_i = \underline{x}_i$, and therefore $\underline{Y} = \underline{X}$. Since $\underline{X}$ is needed to determine $\underline{h}_1^t$, the vectors $\underline{y}_i$ will be determined at this step before (3.2) is executed.

Comparison of (2.19) with (2.17) in this case where $\underline{Y} = \underline{X}$ shows that $\underline{T} \neq \underline{C}$ if $\underline{y}_0 \neq 0$. This situation would usually occur if $\underline{M} \neq \underline{A}$ were chosen to introduce a certain representation for the polynomials t_i of the Euclidean algorithm or if $\underline{M}$ were an approximation to either $\underline{A}$ or $\underline{T}$, as can happen in iterative procedures [1]. However, the algorithm includes the similarity transformation of (2.17) as a special case, as is illustrated in the following example.

B. EXAMPLE

To illustrate the procedure the characteristic polynomial of the following matrix will be determined:

$$\underline{A} = \begin{bmatrix} -1/2 & 0 & 3 \\ 0 & -1 & -2 \\ -1/4 & 0 & -1/2 \end{bmatrix}. \quad (3.3)$$

$\underline{A}$ is cyclic so the choice of $\underline{M} = \underline{A}$ can be made. Although neither $\underline{e}_1$ nor $\underline{e}_3$ are cyclic vectors for $\underline{M}$, $\underline{e}_2$ is a cyclic vector. To avoid fractions the choice $\underline{v} = \underline{y}_3 = 2\underline{e}_2$ will be made. The required sequence of vectors for use in (3.2) and for obtaining $\underline{h}_1^t$ is

$$\begin{bmatrix} \underline{y}_3 \underline{M}^3 \\ \underline{y}_3 \underline{M}^2 \\ \underline{y}_3 \underline{M} \\ \underline{y}_3 \end{bmatrix} = \begin{bmatrix} -2 & -2 & -4 \\ 1 & 2 & 6 \\ 0 & -2 & -4 \\ 0 & 2 & 0 \end{bmatrix} = \underline{Y}. \quad (3.4)$$

Since $\underline{X} = \underline{Y}$, $\underline{h}_1^t$ is the first column of $\underline{Y}^{-1}$ which can be seen to be $\underline{e}_1^t$. A systematic procedure for finding $\underline{h}_1^t$ in the general case will be illustrated in the next section. It is convenient to form the column

vectors $A^{j-1}h_{-1}^t$ in (2.26) in advance. Denoting them as h_{-j}^t , they can be shown in matrix form as follows:

$$\begin{bmatrix} h_{-1}^t & h_{-2}^t & h_{-3}^t \end{bmatrix} = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & 0 & 1/2 \\ 0 & -1/4 & 1/4 \end{bmatrix}. \quad (3.5)$$

The scheme in (3.2) becomes

$$\begin{array}{c} \begin{bmatrix} 0 & 2 & 0 \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{ccc} 2 & 2 & 1 \\ \hline \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} -2 & -2 & -4 \end{bmatrix} \\ \hline \begin{bmatrix} 2 & 2 & 4 \end{bmatrix} \\ \begin{bmatrix} 2 & 4 & 12 \end{bmatrix} \\ \hline \begin{bmatrix} 0 & -2 & -8 \end{bmatrix} \\ \begin{bmatrix} 0 & -4 & -8 \end{bmatrix} \\ \hline \begin{bmatrix} 0 & 2 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 2 & 0 \end{bmatrix} \\ \hline \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \end{array} \end{array} \begin{array}{l} \times h_{-1}^t = 2 \\ \times h_{-2}^t = 2 \\ \times h_{-3}^t = 1 \end{array} \quad (3.6)$$

C. The Krylov and Hessenberg Procedures

If (2.19) is postmultiplied by $\underline{Y}$, the first row becomes

$$\underline{y}_1^M = \underline{e}_1^T \underline{T} \underline{Y} + \underline{y}_0. \quad (3.7)$$

In the case at hand $\underline{T} = \underline{D}_0$, a companion matrix, and $\underline{y}_1^M = \underline{y}_n^M$. Equation (3.7) thus gives

$$\begin{bmatrix} \delta_{11} & \delta_{12} & . & . & \delta_{1n} \end{bmatrix} \underline{Y} = \underline{y}_0 - \underline{y}_n^M. \quad (3.8)$$

Equating columns on both sides gives a system of equations which can be solved for the δ_{1j} . In the special case in which $\underline{y}_0 = 0$, this procedure is the Krylov procedure as it is often described [3]. The Krylov procedure and the algorithm of this section both require the formation of the sequence $\left\{ \underline{y}_n^M \right\}$, but they differ in the manner in which the δ_{1j} are determined.

The relationship with the Hessenberg procedure can be seen by observing a particular method of finding h_{-1}^t , the first column of $\underline{X}^{-1}$. The

method is to factor $\underline{X}$ as follows:

$$\underline{X} = \underline{U}\underline{L}\underline{P}, \quad (3.9)$$

where $\underline{U}$ is unit-upper-triangular, $\underline{L}$ is lower triangular, and $\underline{P}$ is a permutation matrix. The method of factoring is a variation of Gaussian elimination and will be described in terms of the preceding example in which $\underline{X} = \underline{Y}$. The factorization is accomplished in $n-1$ steps, and associated with each step is a unit-upper-triangular matrix $\underline{U}_1^{-1}$ such that

$$\underline{U}^{-1} = \underline{U}_{n-1}^{-1} \cdot \cdot \cdot \underline{U}_2^{-1} \underline{U}_1^{-1}. \quad (3.10)$$

These matrices premultiply $\underline{X}$ in (3.9) to give $\underline{L}\underline{P}$. For the example,

$$\begin{aligned} \underline{U}_2^{-1} \underline{U}_1^{-1} \underline{X} &= \begin{bmatrix} 0 & -1 & 3/2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 6 \\ 0 & -2 & -4 \\ 0 & 2 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -4 \\ 0 & 2 & 0 \end{bmatrix} = \underline{L}\underline{P}. \end{aligned} \quad (3.11)$$

The first step of the factorization procedure is to observe the entry of x_n which has maximum magnitude, in this case x_{32} . $\underline{U}_1^{-1}$ is chosen so that entry (2,2) is zero in $\underline{U}_1^{-1}\underline{X}$, and this product is formed. Similarly, the entry with maximum magnitude in row $n-1$ of $\underline{U}_1^{-1}\underline{X}$ is determined, in this case entry (2,3), and $\underline{U}_2^{-1}$ is chosen so that both entries (1,2) and (1,3) are zero in $\underline{U}_2^{-1}\underline{U}_1^{-1}\underline{X}$. Because of the increasing number of zeros in each row that are introduced, it is seen that $\underline{U}^{-1}\underline{X}$ will be a permutation of a lower triangular matrix, i.e. $\underline{U}^{-1}\underline{X} = \underline{L}\underline{P}$.

When the factorization (3.9) is known, h_1^t is easily determined since

$$h_1^t \equiv \underline{X}^{-1} e_1^t = \underline{P}^{-1} \underline{L}^{-1} \underline{U}^{-1} e_1^t = \underline{P}^{-1} \underline{L}^{-1} e_1^t, \quad (3.12)$$

because $\underline{U}^{-1}$ is unit-upper-triangular. Thus h_1^t can be found by solving

$$(\underline{P}\underline{L})h_1^t = e_1^t \quad (3.13)$$

by back substitution. The factorization (3.9) is also useful in solving (3.8), where $\underline{Y} = \underline{X}$.

If (3.9) were introduced into (2.17), it would be seen that a transformation of $\underline{M}$ to unit-upper-Hessenberg form is implicit. The substitution yields

$$\underline{U}(\underline{L}\underline{P}\underline{M}\underline{P}^{-1}\underline{L}^{-1})\underline{U}^{-1} \equiv \underline{U}\underline{M}'\underline{U}^{-1} = \underline{C}, \quad (3.14)$$

and $\underline{M}'$ must be unit-upper-Hessenberg because $\underline{C}$ is and $\underline{U}$ and $\underline{U}^{-1}$ are unit-upper-triangular.

An alternate approach to finding the characteristic polynomial is to explicitly determine $\underline{M}'$. This can be conveniently done in much the same way as the factorization is performed [3]. The algorithm can then be executed with $\underline{y}_n = \underline{e}_n$ and $\underline{h}_j^t = \frac{\underline{e}_j^t}{\underline{t}_j}$ to obtain characteristic polynomial. (A more general definition of $\underline{h}_j^t$ than previously given includes this choice [1].) This procedure is distinct from the procedure of finding the δ_{1j} from $\underline{M}'$ by evaluating principal minors of $|\underline{s}\underline{I} - \underline{M}'|[1,2,3]$, and is more general since $\underline{y}_0$ is not necessarily zero.

In conclusion, it is seen that the procedure of this section is related to both the Krylov and Hessenberg procedures, but that the determination of the coefficients by "vector division" in (3.2) is distinct from their procedures.

IV. REFERENCES

- [1] T. G. Marshall, Jr., "Tridiagonalization and related matrix and network algorithms for a polynomial space," Proc. Allerton Conf., pp. 544-556, October 1967.
- [2] T. G. Marshall, Jr., "Synthesis of RLC ladder networks by matrix tridiagonalization," IEEE Trans. on Circuit Theory, to appear.
- [3] J. H. Wilkinson, The Algebraic Eigenvalue Problem.
London: Oxford University Press, 1965. Chapt. 4,6.

GENERATION OF EQUIVALENT DC ELECTRONIC CIRCUITS BY INTERPOLATION, EXTRAPOLATION, AND INTRAPOLATION

E. B. KOZEMCHAK and M. A. MURRAY-LASSO*

Bell Telephone Laboratories, Incorporated, Whippany, New Jersey

ABSTRACT

A new method of generating equivalent networks is introduced. The basis of the method is the 2b-vector of branch voltages and branch currents. The method is capable of controlling the topology and element values of the resulting networks, and yields fruitful results in active as well as passive networks.

Most methods of generating equivalent circuits are applications and generalizations of a matrix method proposed by Cauer and later developed by Howitt^{1,2}. Although the method is quite powerful, perhaps because of its generality, it suffers from certain practical disadvantages such as:

1. It is difficult to control the topology of the resulting network.
2. It is difficult to control the element values of the resulting networks.
3. It is difficult to apply to active circuits containing controlled sources.

In order to generate equivalent circuits which are useful in the design of integrated circuits, it is necessary to overcome these practical difficulties. For example, if an equivalent of a transistor circuit is obtained, one should be able to extract the transistor model from the final circuit and still end with useful element values connected between accessible points in the circuit.

In this paper, a method of generating equivalent networks which departs considerably from Cauer's method is introduced. Starting with two or more parent networks which are equivalent in some sense (the same transfer impedance, for example), the method generates equivalent networks with topologies which are combinations of the parent networks. If the parent networks have the same topology, the offspring networks will retain that topology.

In the case of interpolation, sufficient conditions are given for positive element offspring networks, given positive element parent networks. These conditions can be verified by inspection of the parent networks. Extrapolation may start with a set of non-realizable parent networks and end with a realizable offspring network. Intrappolation requires only one parent network. The offsprings are equivalent networks with the same topology but with different types of elements in the branches. All three methods are simple to use and easy to implement on a computer.

Interpolation and extrapolation proceed by taking two or more networks realizing some circuit function; that is, achieving some particular set of

* Presently with the System Engineering Division and System Research Center, Case Western Reserve University, Cleveland, Ohio.

branch voltages and currents when properly excited. The networks are first imbedded in a common topology. This is achieved by adding new branches which may be open or short circuits in the separate parent networks. This is illustrated in Figure 1. The parent networks of (a) and (b) are imbedded into the common topology of (c).

Next, all the branches voltages and branch currents are calculated and arranged in a $2b$ -vector X

$$X = \begin{bmatrix} V_b \\ I_b \end{bmatrix}$$

where V_b is a b -vector of branch voltages and I_b is a b -vector of branch currents. For interpolation, convex linear combinations of the X vectors of each parent network are formed

$$X = a_1 X_1 + a_2 X_2 + \dots + a_n X_n \quad (1)$$

where X_i is the X vector of the i^{th} parent network. The a_i 's are free for the designer to choose, but subject to the condition

$$a_1 + a_2 + \dots + a_n = 1 \quad (2)$$

$$a_1, a_2, \dots, a_n \geq 0 \quad (3)$$

(Often $n = 2$, for which we are interpolating between two networks.)

The vector X in Eq. (1) is the vector of branch voltages and branch currents of the offspring network. An infinity of them may be generated by choosing different a 's.

The realization of the offspring network after the X vector is determined is simple. Resistive branches are determined by the ratio of their branch voltage to branch current. Controlled sources are fixed in value by the ratio of the appropriate branch variables.

That the offspring network must obey Kirchhoff's voltage and current laws is simply proved. The current and voltage laws for each parent network may be written as

$$BV_{b1} = 0, BV_{b2} = 0, \dots, BV_{bn} = 0 \quad (4)$$

and

$$CI_{b1} = 0, CI_{b2} = 0, \dots, CI_{bn} = 0 \quad (5)$$

where C is a cutset matrix for the linear graph of the general topology, and B is a loopset matrix for the same graph. I_{bi} and V_{bi} are the branch current and branch voltage vectors respectively. For the offspring network

$$X = \frac{V_b}{I_b} = a_1 \frac{V_{b1}}{I_{b1}} + a_2 \frac{V_{b2}}{I_{b2}} + \dots + a_n \frac{V_{bn}}{I_{bn}} \quad (6)$$

Thus,

$$BV_b = a_1 BV_{b1} + a_2 BV_{b2} + \dots + a_n BV_{bn} = 0 \quad (7)$$

and

$$CI_b = a_1 CI_{b1} + a_2 CI_{b2} + \dots + a_n CI_{bn} = 0 \quad (8)$$

The offspring networks, therefore, will obey Kirchhoff's laws provided the parent networks do.

In the case of interpolation, it is possible to place restrictions on the parent networks to guarantee positive elements offspring networks. The sufficient conditions for positive element offspring networks are

1. The parent networks have positive elements.
2. The $\frac{V}{I}$ vectors for each of the parent networks lie in the same orthant of $\frac{V}{I}$ space.

Let us first examine the geometrical meaning of these conditions. Condition 1 implies the following for each branch: call the two branch variables whose ratio defines that branch value x and y . In the case of a resistance $x = I_b$ and $y = V_b$. Since the ratio y/x is positive in each parent network, the projection of the X vector onto the x - y plane must lie in the first or third quadrants. If the projection of a second parent network lies in the same quadrant of the x - y plane, as do points A and B in Figure 2, then any interpolation between those points will also be in that quadrant and a positive element value will result. If the parent networks lie in different quadrants, as do points C and D, then interpolation between those points can lead to a quadrant where the element value ($=y/x$) is negative.

The sufficiency of conditions 1 and 2 is easily proved. We are given all parent networks in the same orthant of $\frac{V}{I}$ space, an orthant in which all element values are positive. All offspring networks lie within the convex hull formed by parent networks. But since an orthant is also convex, all points in the convex hull of offspring networks must lie in the same orthant and, therefore, have positive element values.

Condition 2 is not a necessary condition as illustrated by Figure 2. Parent networks at E and F, although in different orthants, yield positive element offspring network. Thus two parent networks may lie in different orthants, and, provided the line joining them goes through the origin, yield positive element offspring networks.

We now show that a transfer impedance Z_{ij} (or driving point impedance Z_{ii}) remains invariant when interpolating from parent networks having equal transfer impedances. Assume a set of parent networks N_k , $k = 1, \dots, N$ excited with a single independent current source labeled branch j of value

1 amp. The transfer impedance Z_{ij} (with second port across branch j) is equal for all the networks.

$$z = (Z_{ij})_k = (V_i)_k, \quad k = 1, \dots, n \quad (8')$$

After interpolation, a current source of 1 amp appears as branch j since the current of branch j is given by

$$I_j = a_1 \cdot 1 + a_2 \cdot 1 + \dots + a_n \cdot 1 = 1$$

according to (2).

The voltage across branch i is

$$V_i = a_1 V_{i1} + a_2 V_{i2} + \dots + a_n V_{in}$$

Since all the voltages V_{ik} are equal

$$V_i = (a_1 + a_2 + \dots + a_n) V_i = z$$

using (7) and (8').

Hence the transfer impedance remains invariant.

Several transfer impedances from the same excitation may remain invariant, hence a whole column of a Z matrix may be kept invariant. A similar proof holds for transfer admittances. The proof is also valid for extrapolation.

To illustrate these concepts, consider the networks of Figures 3a and 3b. Both have a driving point impedance of 1Ω . In order that the two circuits have the same topology, the circuits are augmented as in Figures 4a and 4b. The node voltages of the circuits and the branch currents are indicated on the schematics. We form the vectors.

$$\begin{aligned} X_a^t &= \begin{bmatrix} V_{1a} & V_{2a} & V_{3a} & V_{4a} & V_{5a} & I_{1a} & I_{2a} & I_{3a} & I_{4a} & I_{5a} \end{bmatrix} \\ &= \begin{bmatrix} -1 & 1 & .5 & .5 & .5 & 1 & 0 & 1 & .5 & .5 \end{bmatrix} \end{aligned} \quad (9)$$

$$\begin{aligned} X_b^t &= \begin{bmatrix} V_{1b} & V_{2b} & V_{3b} & V_{4b} & V_{5b} & I_{1b} & I_{2b} & I_{3b} & I_{4b} & I_{5b} \end{bmatrix} \\ &= \begin{bmatrix} -1 & 1 & .5 & .5 & .5 & 1 & .5 & .5 & .5 & 0 \end{bmatrix} \end{aligned} \quad (10)$$

The halfway interpolation ($a_1 = a_2 = 1/2$ in Eq. (1)) is

$$\begin{aligned} X_0 &= \begin{bmatrix} V_{10} & V_{20} & V_{30} & V_{40} & V_{50} & I_{10} & I_{20} & I_{30} & I_{40} & I_{50} \end{bmatrix} \\ &= \begin{bmatrix} -1 & 1 & .5 & .5 & .5 & 1 & .25 & .75 & .5 & .25 \end{bmatrix} \end{aligned} \quad (11)$$

This leads to the offspring network of Figure 5. The equivalence of the offspring network to the parent network is maintained as the driving point impedance seen by the current source remains 1 ohm.

The extrapolation of equivalent networks from two or more parent networks proceeds as in the interpolation. The requirement that positive weights (3) be used is removed. One may thus begin with parent networks employing negative elements (G and H in Figure 2), and extrapolate to a network with positive elements (I in Figure 2).

An equivalent offspring network may be extrapolated from the parent networks of Figures 3a and 3b. Choosing $a_1 = 2$ and $a_2 = -1$ results in an X vector for the offspring network

$$x_0^t = \begin{bmatrix} -1 & 1 & .5 & .5 & .5 & 1 & -.5 & 1.5 & .5 & 1 \end{bmatrix} \quad (12)$$

The offspring network resulting from extrapolation is shown in Figure 6. Note the presence of a negative resistance, indicating the extrapolation leads into a different orthant of $\begin{bmatrix} V \\ I \end{bmatrix}$ space from the orthant of the two parent networks.

Intrapolation consists of reinterpreting the vector X for a single parent network. Thus a resistive network with a particular transfer function may be converted into a network with controlled sources and resistances. For example, assume the network of Figure 5 to be a parent network with

$$z_{11} = \left. \frac{V_1}{I_1} \right|_{I_2=0} = 1 \quad \text{and} \quad z_{21} = \left. \frac{V_2}{I_1} \right|_{I_2=0} = 1/2.$$

By the process of intrapolation, branch 4 is replaced by a voltage controlled current source, the controlling voltage being the voltage across branch 2. The value of g_m for the source is found from (11)

$$g_m = \frac{I_{40}}{V_{20}} = \frac{.5}{1} = .5$$

The equivalent network of Figure 7 results. Notice that one could now begin with an active network (Figure 7) and a passive network (Figures 3a or 3b) as parent networks and generate equivalent offspring networks.

The concept of equivalent circuits by interpolation is extremely useful in the biasing of electronic circuits. Any network which provides the proper bias (proper V_{CE} and I_C) for each of the transistors is equivalent. From among the equivalent bias circuits one can then choose that one which optimizes some other characteristic. Desired ac performance and/or dc stability may be included as an objective in selecting from among the equivalent bias circuits.

An example illustrating this procedure is shown in Figure 8. The requirements on the bias of the differential amplifier are that

1. $I_{C1} = I_{C3} = 7.5 \text{ ma.}$
2. $V_{CE1} = V_{CE3}.$

R_2 , R_3 , R_5 , and R_7 were left variable and two parent networks were generated using a computer program based on Constrained Singular Imbedding^{3,4}. Only the pertinent branch variables are shown below. Those not shown correspond to elements which are invariant. An element that is invariant in all parent networks remains invariant in all offspring networks.

For the first parent network

$$\begin{bmatrix} V_1^t \end{bmatrix} = \begin{bmatrix} V_{b2} & V_{b3} & V_{b5} & V_{b7} & V_{CE1} & V_{CE3} \end{bmatrix} \\ = \begin{bmatrix} 14.780 & 7.2203 & 6.5994 & 4.4780 & 8.8234 & 8.8234 \end{bmatrix}$$

and

$$\begin{bmatrix} I_1^t \end{bmatrix} = \begin{bmatrix} I_{b2} & I_{b3} & I_{b5} & I_{b7} & I_{C1} & I_{C2} \end{bmatrix} \\ = \begin{bmatrix} 7.3652 & 7.3373 & 7.4693 & 4.7070 & 7.5000 & 7.5000 \end{bmatrix}$$

where branch voltages are expressed in volts and branch currents in milliamperes. The corresponding resistance values for this parent network are

$$R_2 = 2,000.$$

$$R_3 = 984.06$$

$$R_5 = 876.92$$

$$R_7 = 945.81$$

For the second parent network

$$\begin{bmatrix} V_2^t \end{bmatrix} = \begin{bmatrix} 14.844 & 7.1561 & 6.7278 & 4.4780 & 8.8234 & 8.8234 \end{bmatrix} \\ \begin{bmatrix} I_2^t \end{bmatrix} = \begin{bmatrix} 71.589 & 71.561 & 7.4653 & 4.7070 & 7.5000 & 7.5000 \end{bmatrix}$$

and the corresponding resistances are

$$R_2 = 204.86$$

$$R_3 = 100.00$$

$$R_5 = 876.92$$

$$R_7 = 932.16$$

The halfway interpolation between these networks is

$$\begin{bmatrix} V_0^t \end{bmatrix} = \begin{bmatrix} 14.812 & 7.1882 & 6.6636 & 4.4780 & 8.8234 & 8.8234 \end{bmatrix} \\ \begin{bmatrix} I_0^t \end{bmatrix} = \begin{bmatrix} 39.477 & 39.449 & 7.4693 & 4.7070 & 7.5000 & 7.5000 \end{bmatrix}$$

and the resistances of the offspring network are

$$R_2 = 372.32$$

$$R_3 = 182.21$$

$$R_5 = 876.92$$

$$R_7 = 938.98$$

Note that the transistor operating points $V_{CE1} = V_{CE3} = 8.8234$ and $I_{C1} = I_{C3} = 7.5$ ma are maintained with the resistance values of the offspring network.

The concept of equivalent networks by interpolation, extrapolation, and intrapolation may also be extended to AC circuits. The numerical manipulations performed on DC networks are replaced by manipulations of polynomials in the Laplace transform variable s . As an example, consider the two parent networks of Figure 9a and 9b, both of which realize a driving point impedance $Z = \frac{s+1}{2s+1}$. Using node voltages (rather than branch voltages) and branch currents to interpolate:

$$X'^t = \begin{bmatrix} V'_0 & V'_1 & V'_2 & I'_1 & I'_2 & I'_3 & I'_4 & I'_5 \end{bmatrix} = \begin{bmatrix} \frac{s+1}{2s+1} & \frac{s+1}{2s+1} & \frac{s}{2s+1} & 1 & 1 & \frac{s+1}{2s+1} & \frac{s}{2s+1} & \frac{s}{2s+1} \end{bmatrix} \quad (13)$$

$$X''^t = \begin{bmatrix} V''_0 & V''_1 & V''_2 & I''_1 & I''_2 & I''_3 & I''_4 & I''_5 \end{bmatrix} = \begin{bmatrix} \frac{s+1}{2s+1} & \frac{1/2}{2s+1} & \frac{1/2}{2s+1} & 1 & 1 & \frac{2s}{2s+1} & \frac{1}{2s+1} & \frac{1}{2s+1} \end{bmatrix} \quad (14)$$

The halfway interpolation between X' and X'' is

$$X^t = \frac{1}{2}(X' + X'') =$$

$$\begin{bmatrix} V_0 & V_1 & V_2 & I_1 & I_2 & I_3 & I_4 & I_5 \end{bmatrix} = \begin{bmatrix} \frac{s+1}{2s+1} & \frac{(1/2)(s+3/2)}{2s+1} & 1/4 & 1 & 1 & \frac{(1/2)(3s+1)}{2s+1} & \frac{(1/2)(s+1)}{2s+1} & \frac{(1/2)(s+1)}{2s+1} \end{bmatrix} \quad (15)$$

By taking ratios of branch voltages to branch currents, the five impedance functions are found and synthesized as in Figure 10. Note that an RLC offspring network results from two RC parent networks.

In conclusion, a new approach to the generation of equivalent networks has been demonstrated. The equivalence of the offspring network to the parent networks results from ratios of voltages and currents (driving point and

transfer immittances, for example) that remain invariant under the operations of interpolation, extrapolation, and intrapolation. These operations are performed on vectors of branch (or node) voltages and branch currents for the parent networks. Interpolation allows control of element values in the offspring network. Since the designer has the option of choosing parent networks, he can exercise control over topology. Generation of parent networks is considered elsewhere^{3,4}. The formulation allows active as well as passive parent and offspring networks.

BIBLIOGRAPHY

1. W. Cauer, "Vierpole," Elek. Nachr. Tech., Vol. 6, pp. 272-282, 1929.
2. N. Howitt, "Equivalent Electrical Networks," Proc. IRE, Vol. 20, pp. 1042-1051, 1932.
3. E. B. Kozemchak and M. A. Murray-Lasso, "Computer-Aided Circuit Design by Singular Imbedding," Bell System Technical Journal, January 1969.
4. M. A. Murray-Lasso and E. B. Kozemchak, "Computer-Aided Circuit Design by Constrained Singular Imbedding," IEEE Transactions on Circuit Theory, to be published.

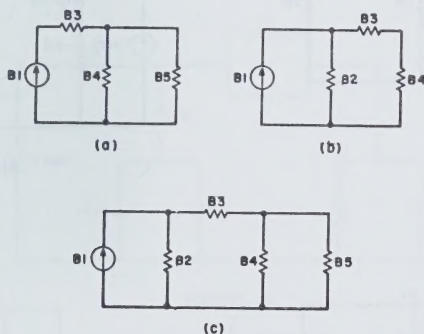


FIGURE 1

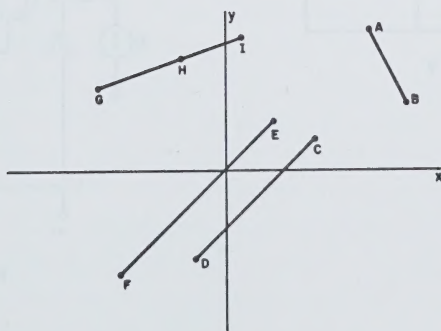


FIGURE 2

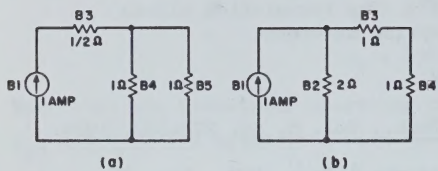


FIGURE 3

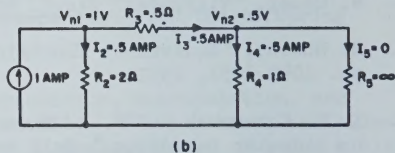
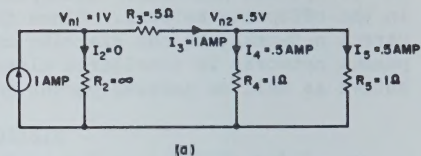


FIGURE 4

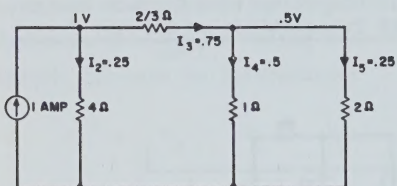


FIGURE 5

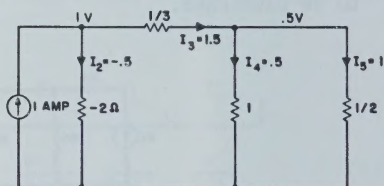


FIGURE 6

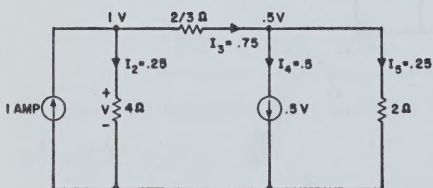


FIGURE 7

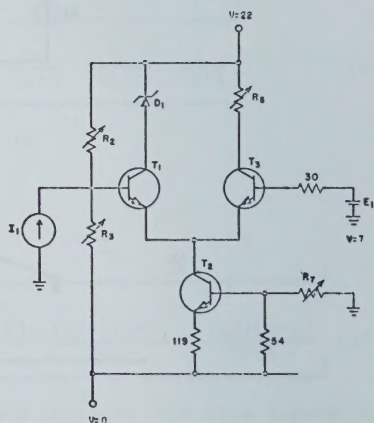
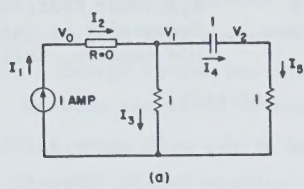
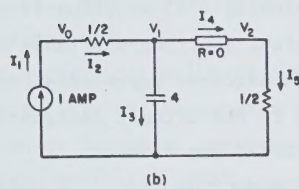


FIGURE 8



(a)



(b)

FIGURE 9

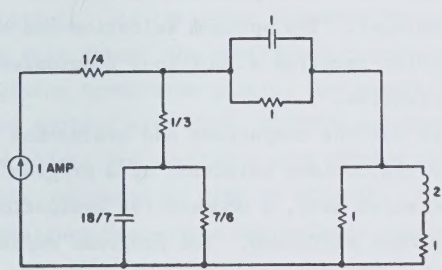


FIGURE 10

CONSIDERATIONS FOR OPTIMUM GCA PROGRAM APPLICATION

DONNAMAIE E. MEYERHOFF
Member of the Technical Staff
Northrop Corporate Laboratories
Hawthorne, California

LAWRENCE P. McNAMEE
Assistant Professor
University of California
Los Angeles, California

ABSTRACT

This paper presents some of the more important factors that should be considered when selecting an optimum General Circuit Analysis program for a given circuit problem, including: 1) modeling flexibility; 2) analysis capability; and 3) the numerical routines available with the programs. The programs emphasized are considered representative of the classes of programs presently available to the circuit designer/analyst. The programs are: ECAP, CIRCUS, SCEPTRE and NASAP.

INTRODUCTION

There are a wide variety of general purpose electronic circuit analysis (GCA) programs available to the potential user, each with its own advantages and disadvantages. The optimum selection and application of a program to a given problem requires a judicious compromise of program capabilities and deficiencies.

To provide a basis for the comparison and evaluation of GCA programs that may be applied to the optimum selection of a program for application by a nonprogramming-oriented user, a comparative evaluation of several major GCA programs has been performed. The programs emphasized in the evaluation are: 1) ECAP, the IBM-360 version and the TRW-ECAP/7094; 2) CIRCUS, developed by Boeing; 3) SCEPTRE, developed by IBM; and 4) NASAP, both the original ERC version (NASA Engineering Research Center, Massachusetts) and the version under development at UCLA. The evaluation consists of: 1) a discussion of the programs as they have been documented in the user's manuals and technical journals; and 2) a discussion of the individual performance of the programs with several sample circuit analysis problems. Although several specific programs are discussed, the basis of program selection formulated may be extrapolated to other programs in existence or as they are developed. The following discussion presents some of the results of this evaluation.

FACTORS INFLUENCING PROGRAM SELECTION

Program Availability. The first area of concern for the potential user should be the availability of the programs to be considered for possible application. The programs to be considered should either be operational or be sufficiently compatible with the installation to allow

fast adaptation. (See Table 1.)

The mode of operation of the program is also a factor. Whether or not a user requires an on-line program depends upon his problem. Off-line operation, or batch processing, is best suited to lengthy computations such as those involved in a design review provided the turn-around time delay is not excessive. On-line operation is best suited for the experimental problem, such as that involved in the actual design process, where the circuit design and the computer equivalent circuit are not preset and the analysis is not defined. Most installations are restricted to the off-line operation of the programs.

Analysis Capability. The application of a GCA program to an analysis is dependent upon the capability of the program for the analysis. CIRCUS and SCEPTRE are restricted to Transient analysis; NASAP and ECAP are capable of DC, AC or Transient analysis. ECAP includes the ability for worst case DC or AC analysis while NASAP provides the user with a transfer function for the circuit. Both programs provide a Bode plot.

Modeling Limitations. An important criteria for program application is that the program will accept the equivalent circuit model desired by the user. When applying transistor models, for example, linear small signal models may be applied with ECAP, SCEPTRE and NASAP. Large signal nonlinear prestored models are available with CIRCUS and SCEPTRE or the user may describe his own model to ECAP or SCEPTRE. The ECAP switch allows a user to construct piece-wise approximate equivalent circuits. (See Table 2.)

If the user desires nonlinear passive elements, ECAP may approximate the elements with a switching model which may be cumbersome, or the user may apply SCEPTRE which has provisions for nonlinear elements to be described directly with FORTRAN equations, or indirectly with tables. SCEPTRE is restricted in the allowable independent variables but more flexible for nonlinear analysis than previous programs have been.

In addition to the semiconductor devices and variable passive elements, the source that is to activate the circuit must also be modeled. The user must consider the program provisions for unusual sources such as oddly shaped pulses, non-periodic repetitive pulses, and provisions for the simulation of radiation-induced current pulses when required. ECAP accepts a maximum of 126 values to describe a periodic or non-periodic pulse, the last value described is held to the end of the run for a non-periodic pulse. ECAP also has a description statement for specifying a sinusoidal source. SCEPTRE will accept an equation descrip-

TABLE 1: PROGRAM - EQUIPMENT COMPATIBILITY

PROGRAM CLASS	ECAP		SCEPTRE*		CIRCUS**		NASAP	
	LINEAR	NONLINEAR (TRANSIENT PROGRAM)	QUASI-LINEAR (TRANSIENT PROGRAM)	LINEAR				
PROGRAM LANGUAGES	FORTAN II-D FORTAN IV-E	FORTAN IV	FORTAN IV	FORTAN IV				
COMPUTERS ON WHICH THE PROGRAM IS PRESENTLY OPERATIONAL	IBM 1620 7090/94 360/75 2250-360 GRAPHIC CDC 3600 6600 PLUS MANY OTHERS	IBM 7090/94 360/75 GE 635 UNIVAC 1108	IBM 7090/94 360/75 GE 635/35 CDC 3400 6600 UNIVAC 1107/1108	IBM 7090/94 360/75/50/40 GE 635/35 CDC 3400 6600 UNIVAC 1107/1108				
COMPUTERS ON WHICH IT IS TO BECOME OPERATIONAL IN THE NEAR FUTURE		IBM 360 HONEYWELL 2200 BURROUGHS 5500	HONEYWELL H-200					
STORAGE REQUIREMENTS	7074 - 32K W/OVERLAY 360 - 131, 072 BYTES GRAPHIC - 256K BYTES	7090/94 - 32K W/OVERLAY 360 - 256K BYTES	7090/94 - 32K W/OVERLAY 360 - 262K BYTES					
ON-LINE CAPABILITY	YES GRAPHIC ECAP	NO	YES TELETYPE	YES TELETYPE				
VERSION REVIEWED IN THIS THESIS	TRW-ECAP/7094 ECAP/360 GRAPHICS ECAP	SCEPTRE/7094	CIRCUS/7094 (EMULATION) CIRCUS/1107	UCLA-NASAP/360 UCLA-NASAP/7094 NASAP-ERC/SIGMA 7				

NOTE: The TRW-ECAP differs from ECAP/360 in plot and list output, double precision arithmetic capacitor modeling and iteration for the matrix inversion.

* Private communication, S. R. Sedore, June 1968.

** Private communication, R. H. Diebhaug, June 1968.

TABLE 2
SUMMARY OF MODEL AND PROGRAM COMPATIBILITY

Program:	NASAP	ECAP	SCEPTRE	CIRCUS	TRAC	NET-1	PREDICT
Model:							
<u>Small Signal</u>							
Low Frequency							
Hybrid	E	E	P	N	N	N	E
Tee	E	E	E	N	N	N	E
High Frequency							
Hybrid-Pi	E	E	E	N	N	N	E
Modified Tee	E	E	E	N	N	N	E
<u>Large Signal</u>							
Ebers-Moll	D	D	P	N	N	P	E
Beaufoy-Sparkes	D	D	E	P	N	N	E
Linville Lumped	D	D	E	N	P	N	E
<u>Radiation-Inclusive</u>							
<u>Large Signal</u>							
Ebers-Moll	N	D	P	N	N	P	E
Beaufoy-Sparkes	N	D	E	P	N	N	E
Linville Lumped	N	D	E	N	P	N	E

Key: E Engineer Derived (Elements available within program are sufficient to allow application of this model.)
P Prestored Model (Available on call from a program library model store.)
D Inefficient (Possible to implement through remodeling; not recommended.)
N Not acceptable (So difficult to implement it is not an approved model for the program.)

tion for a sinusoidal source and a table for pulsed sources. SCEPTRE requires that a repetitive pulsed source be repeatedly described in tabular form until the last time value of the table is equal to the last time value of the solution interval.

CIRCUS has descriptive statements for pulsed and sinusoidal sources, only sinusoidal voltage sources are allowed, and each pulsed source is described by rise and fall times, pulse duration, period, maximum and minimum levels. An oddly shaped pulse requires several sources in series.

Dependent current sources that are linearly dependent and a resistor current are directly described to ECAP, NASAP, and SCEPTRE. Experience has shown that SCEPTRE requires that a tree capacitor be placed in parallel with a linearly dependent current source.

The programs are uniform in the input preparation procedures. All programs require a conventional RLC circuit description; the implementation of a Linville model directly in terms of semiconductor parameters is not presently possible. The programs reviewed are all user-oriented; i.e., a programmer's assistance is not required.

A further consideration when modeling for analysis is the problem size restrictions of the programs. Once a model has been developed it

must be reduced to an acceptable size. Most GCA programs have reasonable problem size limitations.

Numerical Methods. An important factor to be considered in the selection of a program is the numerical method of the program and its effectiveness in relation to the circuit and to expected solution behavior. For example, with transient analysis problems, the accuracy desired in the solution is restricted first by the modeling and second by the accuracy of the integration routine which is a function of the solution waveform. (See Table 3.)

TABLE 3
SUMMARY OF PER-STEP TRUNCATION ERROR FOR THE GCA
PROGRAM INTEGRATION ROUTINES

PROGRAM	PROBLEM FORMULATION METHOD	INTEGRATION ROUTINE	TRUNCATION ERROR-PER-STEP
ECAP/360	NODE EQUATIONS	EXPLICIT NUMERICAL INTEGRATION	$(\Delta t)^3$ - Simulation of L
			$(\Delta t)^2$ - Simulation of C
TRW-ECAP	NODE EQUATIONS	EXPLICIT NUMERICAL INTEGRATION	$(\Delta t)^3$ - Simulation of L and C
CIRCUS	STATE VARIABLES	EXPONENTIAL METHOD	$\approx (\Delta t)^3$ to $(\Delta t)^4$
SCEPTRE	STATE VARIABLES	RUNGA-KUTTA-4th	$(\Delta t)^5$
		EXPONENTIAL METHOD	$\approx (\Delta t)^3$ to $(\Delta t)^4$
		TRAP-low order PREDICTOR-CORRECTOR	$(\Delta t)^3$
NASAP	FLOWGRAPH	FAST FOURIER TRANSFORM	not defined; under investigation

Problems involving a wide range of time constants are best solved by CIRCUS and SCEPTRE using XPO, provided the smallest eigenvalue is real. Problems involving oscillating waves should not be solved with SCEPTRE-TRAP or ECAP/360 if it is possible to use another program. The accuracy of CIRCUS, while the program prints only two places or three if an exponent is required, is seen to be sufficient for most engineering problems.

Run Controls. There is a close relationship between the run controls, such as time step selection, per step allowable error, and the final solution accuracy. ECAP and NASAP hold a fixed user-specified step interval over the solution interval. CIRCUS and SCEPTRE apply variable step

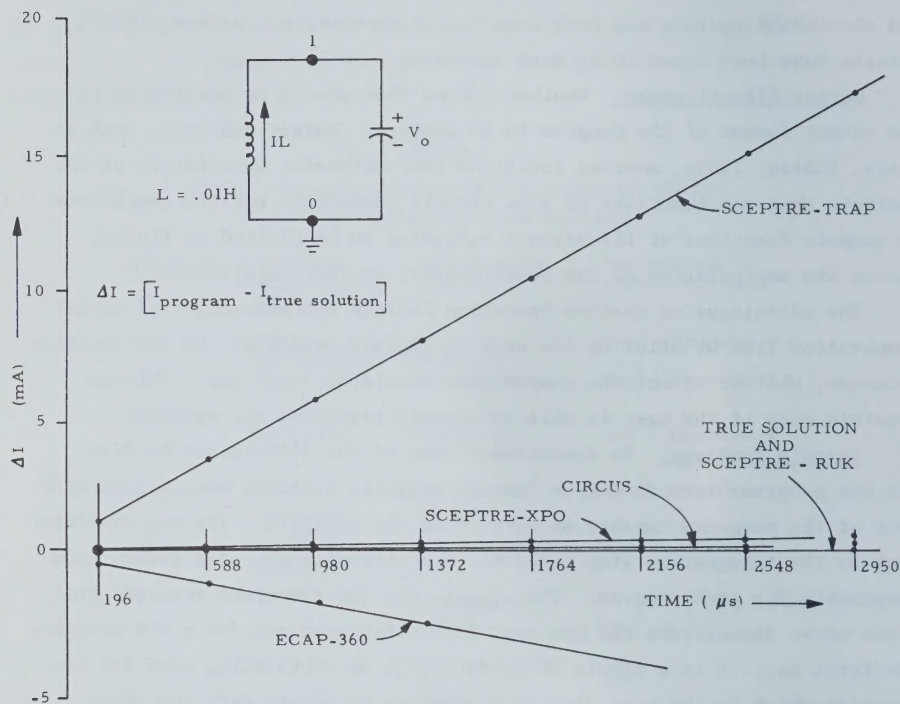
intervals over a solution which the programs compute at each step. SCEPTRE and CIRCUS select Δt from the $\Delta t_{\min}$ to $\Delta t_{\max}$, selection range specified by the user on the basis of error criteria which the user may also control with SCEPTRE. The SCEPTRE program applies a halving-doubling routine to vary step size; the CIRCUS program varies the step according to an error formula criteria. Note that while the XPO routine of SCEPTRE and the CIRCUS routine are both from Pope's exponential method, CIRCUS results have been found to be more accurate.

Output Effectiveness. Another factor that should be considered is the output format of the program to be applied. Output features, such as plots, tables, lists, monitor functions (for automatic termination of the analysis that are functions of some circuit variable), and the capability to compute functions of the circuit variables to be plotted or listed, reduce the manipulation of the results prior to their application.

The advantages of on-line operation include the prevention of wasted computation time by allowing the user to rapidly determine, as the analysis proceeds, whether or not the computation should be continued. This is possible only if the user is able to rapidly interpret the results.

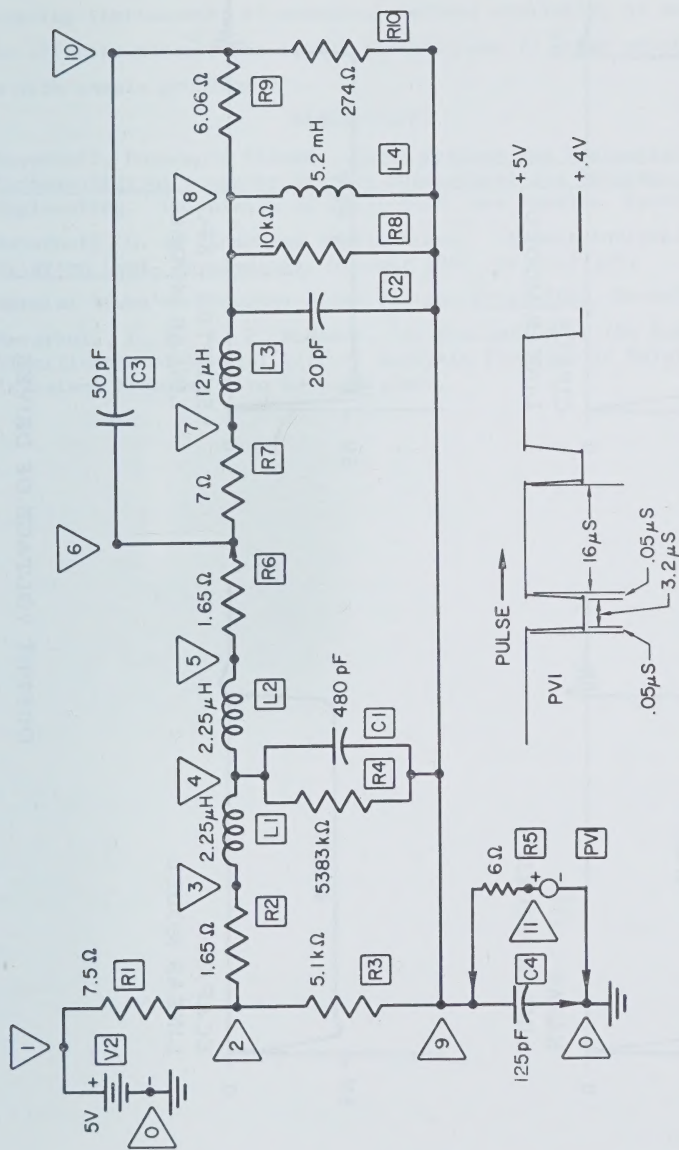
Example Problems. To demonstrate some of the differences between the GCA programs several sample circuit analysis problems were solved with each of the programs capable of performing the analysis. The run controls such as the integration step interval, and allowable per-step error, were comparable for each program. The results for two transient analysis problems shown demonstrate the two most formidable problems for a GCA program. The first circuit is a simple LC circuit with an oscillating wave for its solution which is the most difficult waveform to obtain with the integration routines. (See Figure 1.) The graph shows the error per magnitude peak for each of the programs applied.

A second problem is a nonlinear transformer which was modeled with a standard linear model. (See Figure 2.) The variation in time constants in this problem presents another difficulty for the GCA programs. As the solution waveforms demonstrate, the linear model was sufficient if the user desires the output voltage. (See Figure 3.) Although the ECAP/360 solution did not reflect the oscillations on the pulse, the magnitude level, which was the desired solution, was within the ± 0.25 volt allowable error. (The solution from laboratory measurements is 5.0 volts.) The ECAP waveform could be improved by taking smaller time steps over the solution; however, the gain in accuracy would not justify the cost of the solution.



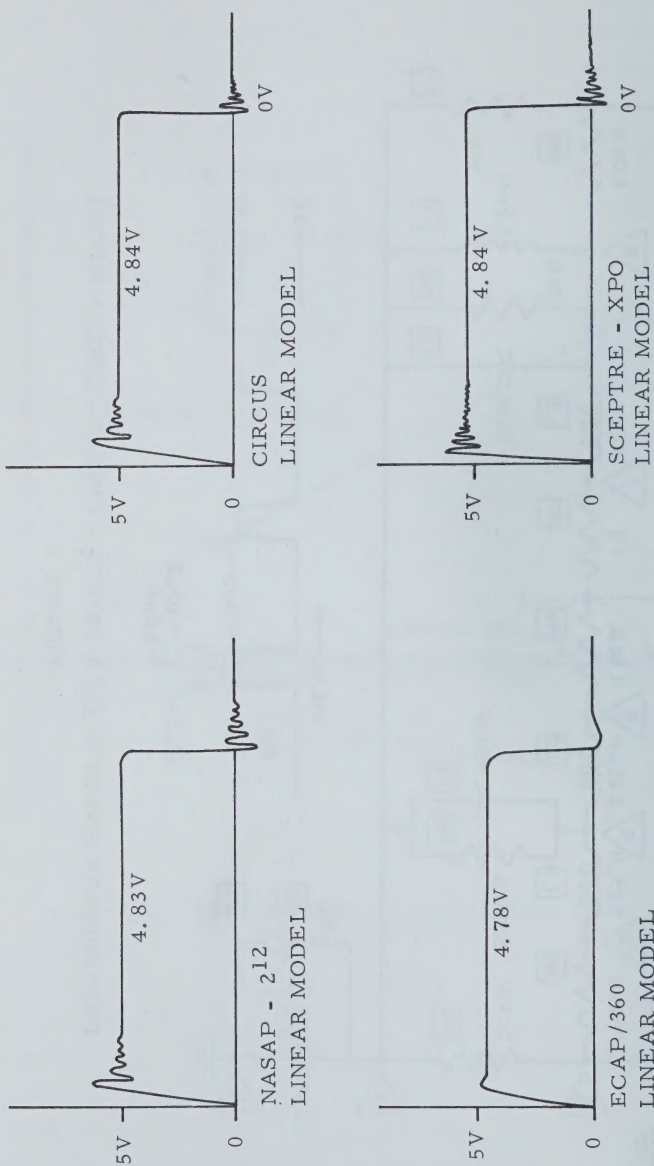
DIVERGENCE OF THE PEAK CURRENT VALUE
VERSUS TIME

Figure 1



PRELIMINARY ANALYSIS OF A DRIVER - CIRCUS - LINEAR MODEL

FIGURE 2



OUTPUT VOLTAGE OF DRIVER

Figure 3

CONCLUSIONS

This paper has highlighted the important areas of consideration in the optimum selection of a GCA program for application to a problem. These areas include: 1) program availability; 2) analysis capability; 3) modeling limitations; 4) numerical methods available; 5) run controls; 6) the effectiveness of the output formats; and 7) known program performance with sample problems.

BIBLIOGRAPHY

1. Meyerhoff, Donna Marie Eileen. Investigation and Evaluation of General Purpose Digital Computer Circuit Design/analysis Programs. M.S. in Engineering. University of California, Los Angeles. December, 1968
2. Meyerhoff, D. E. "Computer Applications: Circuit Analysis". 5th Allerton Conf. Proceedings, October 1967, pp. 513-520.
3. Special Issue on Computer Aided Design, Proc. IEEE, November 1967.
4. Meyerhoff, D. E., L. P. McNamee, "An Evaluation of the Numerical Significance of General Circuit Analysis Programs in Relation to the Transient Problem," (to be published).

"INITIAL CONDITIONS" ALGORITHM FOR STATE VARIABLE ANALYSIS OF LINEAR SYSTEMS

F. H. HERMAN* AND S. D. BEDROSIAN
The Moore School of Electrical Engineering
University of Pennsylvania
Philadelphia, Pennsylvania 19104

ABSTRACT

Analysis of linear systems by state variable techniques may involve difficulties with initial conditions. Algorithms are presented for obtaining directly the initial conditions on one side of the starting time from known values at the other side while properly taking into account any singularities. Computer print-outs for several examples are included.

SUMMARY

To obtain the transient response of a fixed linear system one must in general deal with the initial conditions of the independent variable. For convenience, we note the initial conditions (I.C.) defined for the negative limit from the initial point $t = 0$ as $I.C._-$ and that for the positive limit as $I.C._+$. A discontinuity of the initial conditions such that the value of $I.C._-$ is not the same as $I.C._+$ comes about from the input being a singularity function. (1) Various numerical techniques (2,3) for obtaining the transient response require the $I.C._+$. This avoids the discontinuities and possible impulses at the output and its derivatives. In most problems involving physical systems, only $I.C._-$ are known. To provide a simple general method for obtaining the desired $I.C._+$, we develop algorithms for finding the $I.C._+$ corresponding to the known $I.C._-$.

Since the Laplace transform can handle singularity functions such as the Dirac delta function $\delta(t)$ and its derivatives, it is a logical point of departure. The well known initial-value property for a function $f(t)$ and its Laplace transform $F(s)$ will yield $f(0_+)$ provided the limit $s \rightarrow \infty$ exists for $sF(s)$. If $f(t)$ is itself singular at $t=0$, this limit will not exist.(4) By interpreting the significance of the singularity function, one can extend instead of abandon the initial-value property. In particular if $F(s)$ is a rational function with numerator

$\sum_{i=0}^m b_i s^i$ and denominator $\sum_{j=0}^n a_j s^j$ and $a_n = 1$ then use of the initial-value theorem yields

$$f(0_+) = \lim_{s \rightarrow \infty} sF(s) = \lim_{s \rightarrow \infty} \left[b_m s^{m+1-n} \right]$$

Then

$$f(0_+) = \begin{cases} 0 & m < n-1 \\ b_m & m = n-1 \\ \text{undefined} & m \geq n \end{cases}$$

The conventional interpretation of the initial-value property only applies to the first two cases. (5)

* Presently with U.S. Army Picatinny Arsenal, Dover, New Jersey 07801.

For $m = n$, division of the numerator of $F(s)$ by the denominator yields $F(s) = b + F_1(s)$. Then $f(t) = b \delta(t) + L^{-1}\{F_1(s)\}$ indicating an impulse of magnitude b at the origin. If $m > n$, then $F_1(s)$ would also be singular at the origin.^m By repeated division, the remainder term is reduced until the order of the numerator is less than the order of the denominator. The fraction can then be handled by the initial-value theorem while the remaining terms represent impulses of order equal to the corresponding powers of s . An algorithm, developed for use when $m \geq n$ yields the impulses of $f(t)$ and its derivatives at the origin. To obtain the value of $f(t)$ at $t=0_+$, we note that shortly after time zero, the impulse along with corresponding singularities have passed. For the transient response, $t > 0$, then we disregard impulses and state:

Modified Initial-Value Theorem:

If a. $F(s)$ the Laplace transform of $f(t)$ exists;

b. $\lim_{s \rightarrow \infty} [sF(s)]$ does not exist;

c. $F(s)$ can be put in the rational function form indicated above, then we obtain an expression for $f(\epsilon)$, where ϵ is the positive limit point of the origin, that defines the "modified initial-value" of $f(t)$, i.e., $f(\epsilon) \equiv f(0_+)$. We are also able to determine the modified initial-values of the $n-1$ derivatives of $f(t)$ for an n th order system.

For application to the state variable approach the system equations in the linear normal form are: $\dot{\underline{q}} = \underline{A}\underline{q} + \underline{B}\underline{x}$. Taking the Laplace transform yields: $[\underline{SI} - \underline{A}] \underline{Q}(s) = \underline{B}\underline{X}(s) + \underline{q}(0_-)$ and

$$\underline{Q}(s) = [\underline{SI} - \underline{A}]^{-1} \underline{B}\underline{X}(s) + [\underline{SI} - \underline{A}]^{-1} \underline{q}(0_-).$$

Applying the modified initial-value theorem we get

$$\underline{q}(0_+) = \underline{q}(0_-) + \lim_{s \rightarrow \infty}^* [\underline{SI} - \underline{A}]^{-1} \underline{B}\underline{X}(s).$$

Here the special limit is actually taken separately for each element of the matrix resulting from the expression within the brackets. Several examples are included using the Fortran IV programs of the algorithms.

REFERENCES

1. DeRusso, P. M., Roy, R. J., and Close, C. M.; STATE VARIABLES FOR ENGINEERS, John Wiley & Sons, Inc., New York, 1965, p 11.
2. Liou, M. L., "A Novel Method of Evaluating Transient Response," Proc. IEEE, Vol. 54, January 1966, p 20-23.
3. Valand, J., "Calculation of Transient Response," Electronic Letters, June 28, 1968, p 261.
4. Schwartz, R. J. and Friedland, B.; LINEAR SYSTEMS, McGraw Hill Company, New York, 1965, p 181.
5. Rasof, B., "The Initial and Final Value Theorems in Laplace Transform Theory", J. Franklin Institute, Vol. 274, p 165-177, September, 1962.

MINIMAL VARIANCE FEEDBACK CONTROLLERS: INITIAL STUDIES OF SOLUTIONS AND PROPERTIES

STANLEY R. LIBERTY AND MICHAEL K. SAIN
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana

ABSTRACT

The present paper contains representative quantitative data which were obtained in the initial studies of solutions of equations arising in the optimization of certain stochastic systems according to the criterion of minimizing the cost variance about a fixed mean cost, not necessarily the minimal mean cost. Included are the problem statement, the equations which define the solution, and examples of numerical studies on first-order, second-order, and fifth-order dynamical systems, the latter two being from papers selected from the literature. Restrictions on the length of the paper preclude any significant discussion of the theoretical development of the optimal solution, and these results will be presented elsewhere in more detail.

PROBLEM STATEMENT

Consider the linear stochastic differential controlled object defined by the stochastic differential equation

$$dx(t) = [A(t)x(t) + B(t)u(t)]dt + dw(t) \quad (1)$$

on the time interval $\{t: t_0 \leq t \leq t_F\}$. The state at time t_0 is denoted by x_0 . Let R^n denote the n -fold product of the real line R . Then $x(t) \in R^n$ is the dynamic state, and $u(t) \in R^m$ is the control action. The disturbance process $w(t) \in R^n$ is of Wiener type with nonnegative definite covariance weighting matrix Q given by

$$E\{w(t)w'(s)\} = Q \min[t, s], \quad (2)$$

where the superscript $(')$ denotes transposition and $E\{\cdot\}$ denotes the mathematical expectation. The matrices A and B are assumed continuous in their arguments and of dimension compatible with that of the dynamic state and the control action.

The control action itself is taken to be based upon a linear control law of the form

$$u(t) = k(t, x(t)) = K(t)x(t) \quad (3)$$

where $K(t)$ is to be chosen in connection with the quadratic cost measure given by

$$J(t, x; K) = x'(t_F)P_3x(t_F) + \int_t^{t_F} x'(s)[P_2 + K'(s)P_1K(s)]x(s)ds. \quad (4)$$

P_3 and P_2 are nonnegative definite and P_1 is positive definite. The rather

The development of the theory upon which these quantitative studies are based was carried out at the University of Notre Dame under the sponsorship of the National Science Foundation by means of their grant GK 804.

well known theory of linear stochastic regulators specifies the control law to minimize the mathematical expectation of (4) subject to the measurement of $x(t)$, or, by the separation theorem, subject to suitable noisy measurements of outputs which are linearly related to $x(t)$. Because of the space constraints here, and because the extension to noisy measurements is straightforward, the present discussion is limited to the case in which $x(t)$ can be measured and is directly available for control purposes.

The theoretical extension of this well known theory which is considered herein allows the presetting of the average cost of operation at the value

$$E\{J(t, x; K)\} = M(t, x), \quad (5)$$

which is of course at least as great as the minimum obtained in the usual linear stochastic regulator problem, and then carries out the determination of the control law in such a way that the cost variance

$$V(t, x; K) = E\{J^2(t, x; K)\} - M^2(t, x) \quad (6)$$

is minimized with respect to the class of K satisfying (5). It turns out that this class is non-empty, provided that M is large enough, and is in fact rich enough to permit a meaningful minimization.

The initial results in minimization of cost variance were published in 1965 [1] and development has since continued [2,3,4,5,6]. In addition, two dissertations are in progress: one in discrete-time systems and one of the density of the cost. The importance of the present work is that it is the first serious study of feedback solutions on a significant scale, and the results seem to support continued investigations.

FORM OF SOLUTION

Under the assumption (3), it is a well known fact that a sufficiently general form for (5) is

$$M(t, x) = x^T M(t) x + m(t). \quad (7)$$

It is equally true, but not so well known, that a sufficiently general form for (6) is

$$V(t, x) = x^T V(t) x + v(t), \quad (8)$$

where the notation in (8) is used for the minimum of (6) subject to (5). The functions $m(t)$ and $v(t)$ are easily determined by integrating the traces of $M(t)$ and $V(t)$, so that the latter two matrices become the most interesting quantities in the theory.

It will be shown in a forthcoming paper that the matrices $M(t)$ and $V(t)$ satisfy the pair of simultaneous Riccati equations

$$\begin{aligned} \dot{V}(t) + 4M(t)QM(t) + A^T(t)V(t) + V(t)A(t) - M(t)B(t)P_1^{-1}B^T(t)V(t) + \\ - V(t)B(t)P_1^{-1}B^T(t)M(t) - 4\mu(t)V(t)B(t)P_1^{-1}B^T(t)V(t) = 0 \end{aligned} \quad (9)$$

$$\begin{aligned} \dot{M}(t) + A^T(t)M(t) + M(t)A(t) + P_2 - M(t)B(t)P_1^{-1}B^T(t)M(t) + \\ + \mu(t)V(t)B(t)P_1^{-1}B^T(t)V(t) = 0 \end{aligned} \quad (10)$$

subject to the boundary conditions

$$V(t_F) = 0; \quad M(t_F) = P_3. \quad (11)$$

The nonnegative real scalar function $\mu(t)$ may be prechosen in the solution and is equivalent to the choice of the average cost. For example, as $\mu(t)$ becomes smaller, for a particular t , the average cost becomes smaller, and as $\mu(t)$ becomes larger the average cost becomes larger. The cost variance, on the other hand, has a reciprocal behavior with $\mu(t)$.

BRIEF DESCRIPTION OF DATA

For a more detailed discussion of the figures which follow, the reader is invited to examine reference [6], which is available. Consider now a first order system (1), and the steady state solutions of (9) and (10) for constant μ . Figure 1 shows the variation of the cost variance weighting in root locus form as μ changes (positive real portion of locus), and Figure 2 shows the corresponding locus for the cost mean (finite positive real part of locus). The loci are typical. Figures 3 and 4 show plots of cost mean and variance weighting versus time for two cases in which the tradeoff function $\mu(t)$ is increasing and decreasing, respectively, with time. It is useful to recall that, for a particular t , $\mu(t)$ and cost mean are directly proportional whereas $\mu(t)$ and cost variance are inversely related. Figure 5 shows the corresponding plots for three constant values of μ .

Data was also taken for the motion of a point mass along the real line, as in the published example of Tung [7]. Figure 6 shows the two eigenvalues for mean and variance matrices and the fact that one eigenvalue is dominant. Further plots show only the dominant eigenvalue. Figure 7 shows a crossover effect found in the variance eigenvalue, and apparently closely related to the fact that the open-loop system is not asymptotically stable. Figure 8 shows the more typical nesting of curves for nested values of μ , the effect being produced by asymptotically stabilizing the open-loop system. Figure 9 shows what can be achieved with the unmodified open-loop system by choosing a non-constant function μ . To check the effect of open-loop stability, the matrix A was set equal to zero in the first-order example, and the results showed greater sensitivity to μ , as seen in Figure 10. Figure 11 shows the second-order example modified so that the state matrix of the system has only one pole at the origin instead of two. The effect is in between that of Figures 7 and 8. The effect was also seen to be dependent on P_2 . To illustrate this point, Figures 12, 13, and 14 have been included for the state matrices A and weighting matrices P_2 indicated thereon.

Data was also obtained for the string of high speed moving vehicles problem of Athans and Levine [8]. It essentially corroborated the results of the second-order studies, and so no curves are included here.

REFERENCES

1. M. K. Sain, "On Minimal Variance Control of Linear Systems with Quadratic Loss," Ph.D. Dissertation, University of Illinois, Urbana, January 1965.
2. M. K. Sain, "Relative Costs of Mean and Variance Control for a Class of Linear, Noisy Systems," Proc. Third Allerton Conf. on Circuit and System Theory, pp. 121-129, October 1965.
3. M. K. Sain, "Control of Linear Systems according to the Minimal Variance Criterion--A New Approach to the Disturbance Problem," IEEE Trans. on Automatic Control, vol. AC-11, pp. 118-122, January 1966.
4. M. K. Sain, "A Sufficiency Condition for Minimum Variance Control of Markov Processes," Proc. Fourth Allerton Conf. on Circuit and System Theory, pp. 593-599, October 1966.
5. Luis Cosenza, "Covariance Matrices and Optimal Modes in a Minimum Variance Control Problem," M.S. Thesis, University of Pennsylvania, Philadelphia, May 1967.

6. Stanley R. Liberty, "Minimal Variance Feedback Controllers: Initial Studies of Solutions and Properties," M. S. Thesis, University of Notre Dame, Notre Dame, Indiana, January 1969.
7. F. Tung, "Linear Control Theory Applied to Interplanetary Guidance," IEEE Trans. on Automatic Control, vol. AC-9, pp. 82-84, January 1964.
8. W. S. Levine and M. Athans, "On the Optimal Error Regulation of a String of Moving Vehicles," IEEE Trans. on Automatic Control, vol. AC-11 pp. 355-361, July 1966.

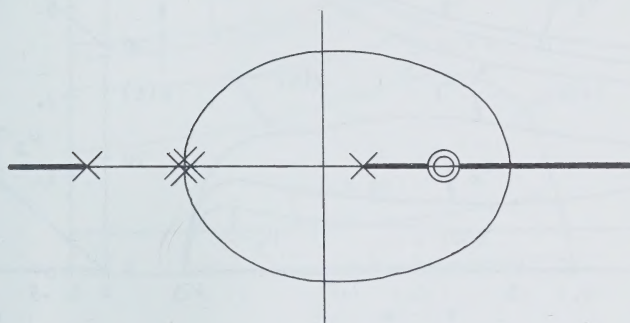


Figure 2. Root Locus Study of the Steady-State Solution of the Mean Cost Equation (10) for $0 < \mu < \infty$. Of Interest Is the Finite Positive Portion of the Locus, Moving from the Real Pole toward the Double Zero as μ Increases Positively.

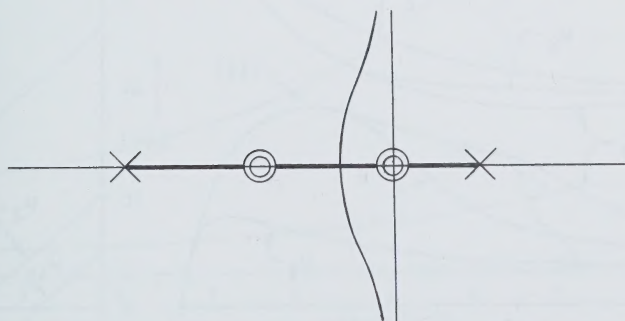


Figure 1. Root Locus Study of the Steady-State Solution of the Cost Variance Equation (9) for $0 < \mu < \infty$. Of Interest Is the Positive Real Portion, Moving from the Positive Real Pole to the Origin as μ Increases Positively.

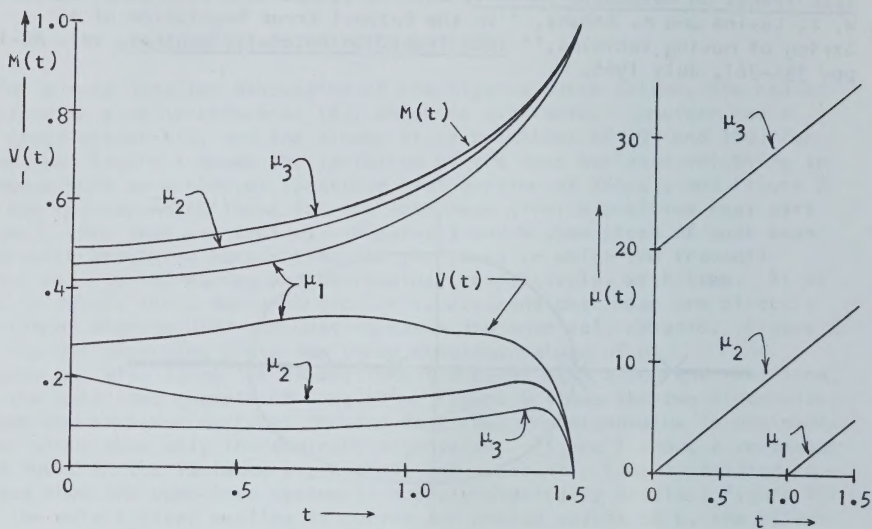


Figure 3. First Order System for $\mu(t)$ Increasing with t .

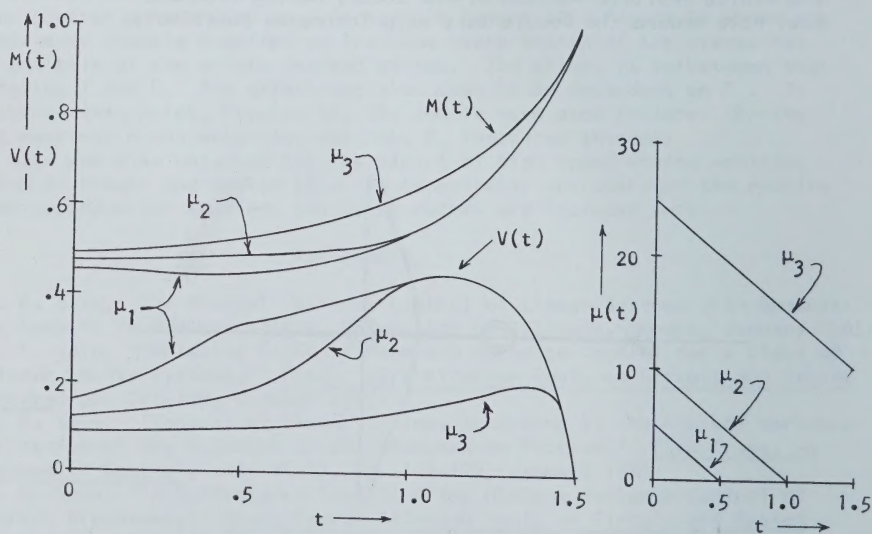


Figure 4. First Order System for $\mu(t)$ Decreasing with t .

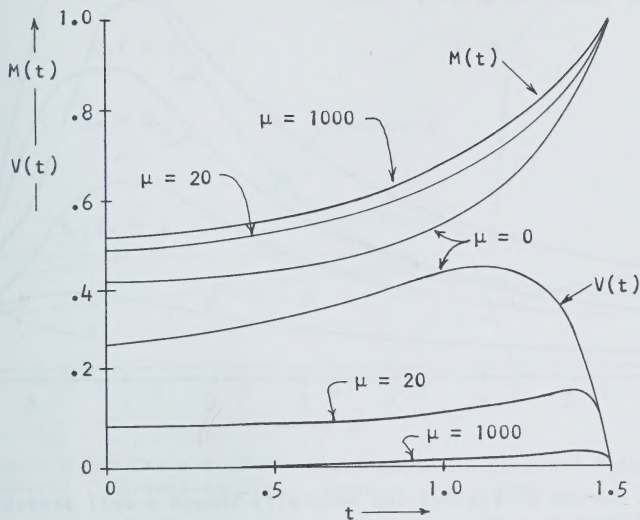


Figure 5. Cost and Variance Weightings: First Order, μ Constant

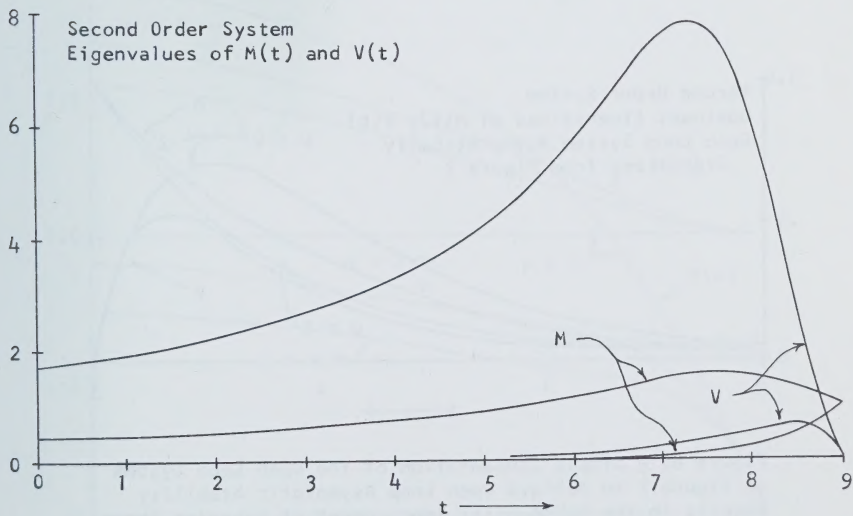


Figure 6. Eigenvalue Behavior of Second Order System: Point Mass

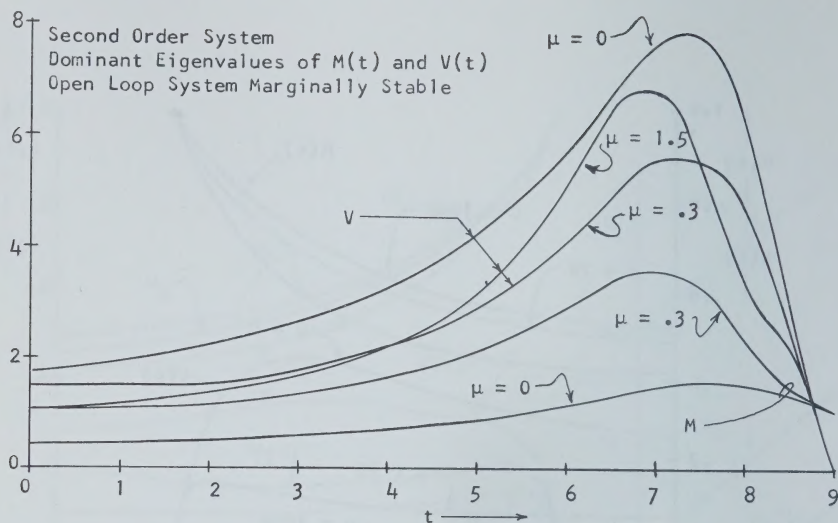


Figure 7. Curves of Eigenvalues Generally Showed a Well Behaved Nesting Effect as μ Varied. When the Open Loop System Was Not Asymptotically Stable, However, Crossovers and Steep Variations Were Found, as Above.

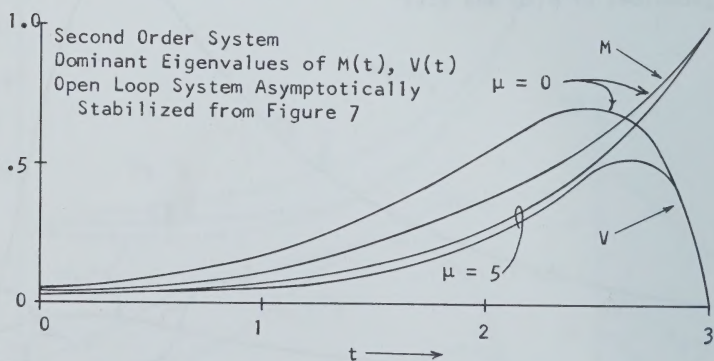


Figure 8. A Simple Compensation of the Open Loop System of Figure 7 to Achieve Open Loop Asymptotic Stability Results in the Substantial Improvement of Behavior Shown.

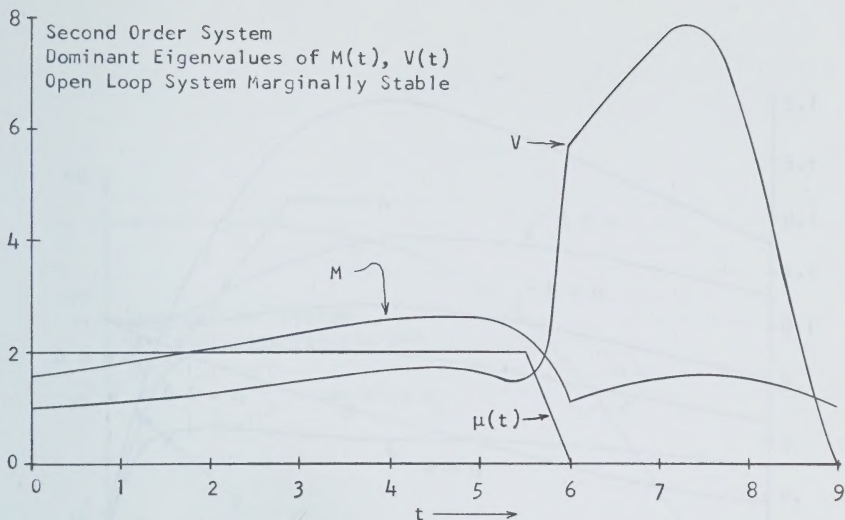


Figure 9. An Example of the Improvement in Variance Behavior by means of Time-Varying $\mu(t)$. Compare with Figure 7.

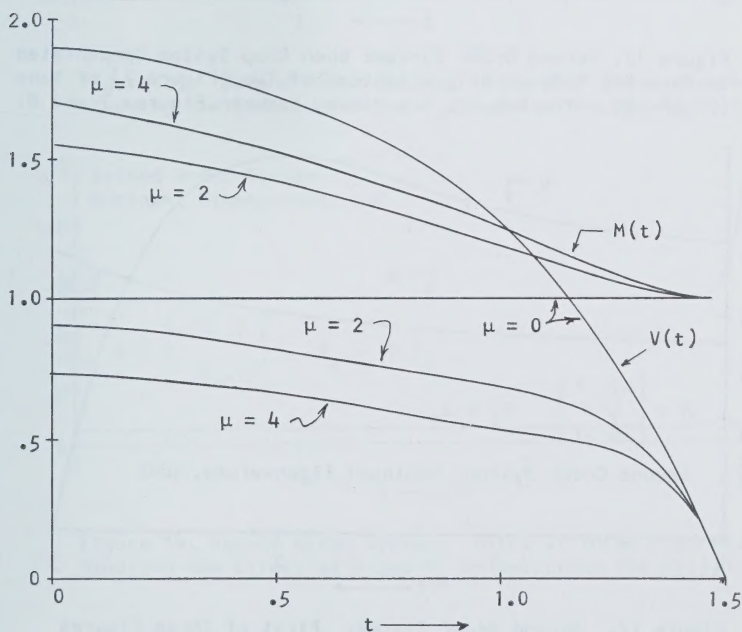


Figure 10. As a Check on the Effects of Marginal Stability, the First Order System Was Re-Examined with $A = 0$. Compare with Figure 5 to Note Increase in Sensitivity to μ .

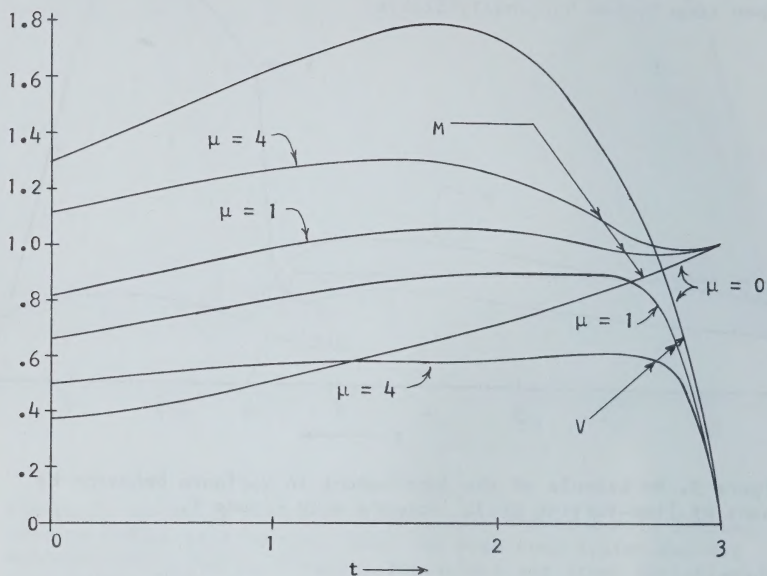


Figure 11. Second Order System: Open Loop System Compensated to Have One Pole at Origin Instead of Two (Figure 7) or None (Figure 8). The Results Are Midway Between Figures 7 and 8.

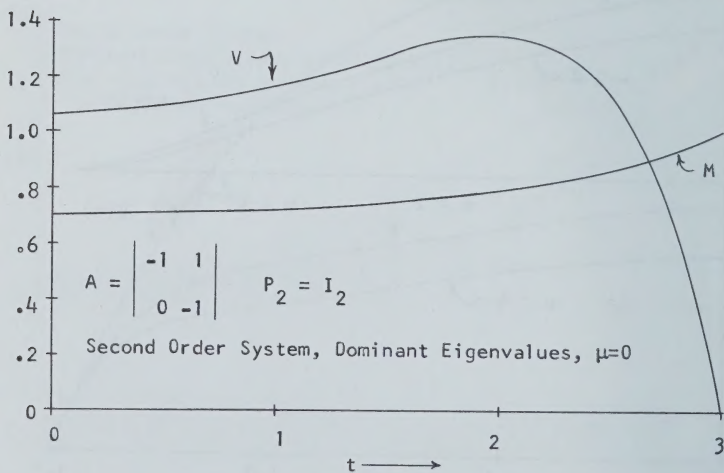


Figure 12. Second Order System. First of Three Figures Studying the Effect of A and P_2 on Solutions for $M(t)$, $V(t)$.

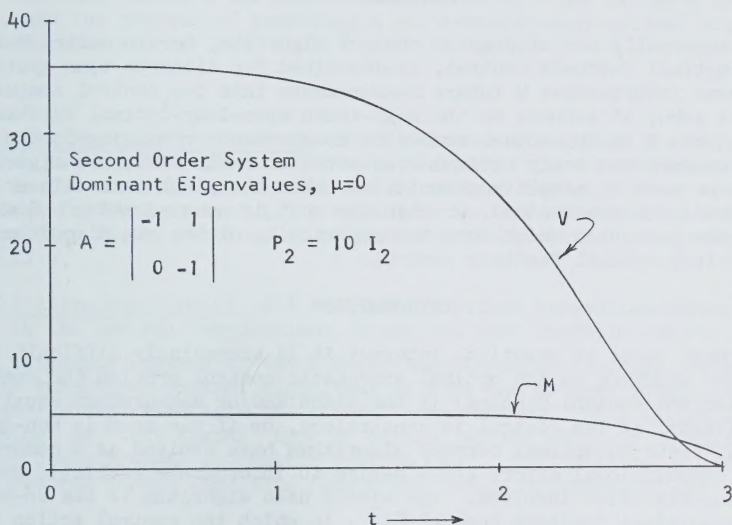


Figure 13. Second Order System. Second of Three Figures Studying the Effect of A and P_2 on Solutions for $M(t)$, $V(t)$.

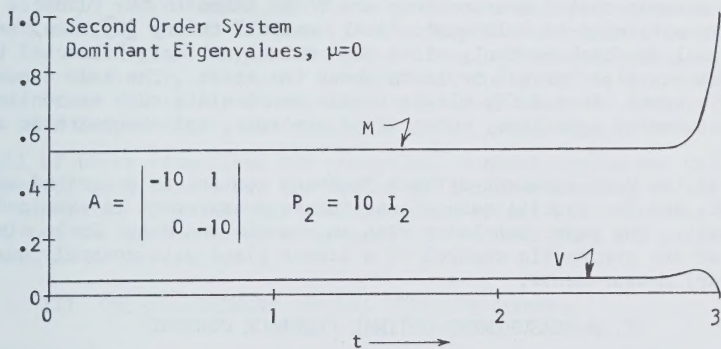


Figure 14. Second Order System. Third of Three Figures Studying the Effect of A and P_2 on Solutions for $M(t)$, $V(t)$.

RENWICK E. CURRY**

Experimental Astronomy Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts

ABSTRACT

An apparently new stochastic control algorithm, herein called M-measurement-optimal feedback control, is described for discrete time systems. This scheme incorporates M future measurements into the control computations: when M is zero, it reduces to the well-known open-loop-optimal feedback control; when M is the actual number of measurements remaining in the problem, it becomes the truly optimal stochastic control. This new algorithm may also be used to simplify computations when the plant is nonlinear, when the controls are constrained, or when the cost is nonquadratic. Simulation results are presented which show the superiority of the new algorithm over the open-loop-optimal feedback control.

I. INTRODUCTION

In most cases of practical interest it is exceedingly difficult to obtain the solution to the optimal stochastic control problem (or combined estimation and control problem) if the plant and/or measurement equations are nonlinear, if the control is constrained, or if the cost is non-quadratic. Different suboptimal control algorithms have evolved as a compromise between computational effort and a desire to incorporate realistic modeling of the uncertainties involved. One widely used algorithm is the so-called open-loop-optimal feedback control [1], in which the control action is computed under the assumption that no measurements are to be taken in the future.

The algorithm presented here, called M-measurement-optimal feedback control, assumes that M measurements are to be taken in the future. This control is more akin to Feldbaum's "dual control" theory [2] than the open-loop-optimal feedback control, since the control not only "directs" the state, but can also "probe" or learn about the state. The same concept can be used to treat other difficult-to-handle constraints such as nonlinear state/measurement equations, constrained controls, and nonquadratic cost criteria.

First the M-measurement-optimal feedback control is described and discussed, and the special case of one future measurement is examined in more detail. The paper concludes with an example and Monte Carlo simulation results of the stochastic control of a linear plant with coarsely quantized (nonlinear) measurements.

II. M-MEASUREMENT-OPTIMAL FEEDBACK CONTROL

Description of the Control Algorithm

The algorithm proceeds as follows: measurements and control actions are recorded up to (say) time t_k , and an estimate of the state vector (and,

* Supported by NASA Grant Nsg 254-62. This work is based on a portion of the author's Ph.D. Dissertation submitted to the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Mass.

** Now with the School of Electrical Engineering, Cornell University, Ithaca, New York.

perhaps, its conditional probability density function) is made. It is temporarily assumed that M measurements are to be taken in the future (at preassigned times), and the control sequence $\{u_k, u_{k+1}, \dots\}$ is computed to minimize the cost to complete the process taking M measurements along the way. This control sequence is the M-measurement-optimal control sequence. Only u_k , the first member of the sequence, is actually implemented; this advances the time index, a new measurement is taken, a new estimate is computed, and the process of computing a new M-measurement-optimal control sequence is repeated.

Remarks

Limiting Forms of the Control The M-measurement-optimal feedback control has two limiting forms: when M is zero, no future measurements are included in the control computations and it becomes the well-known open-loop-optimal feedback control [1]; when M is the actual number of measurements remaining in the problem, it becomes the truly optimal stochastic control [2].

Qualitative Description All available past and present measurements are used in the control computations, hence the name feedback control.

The primary advantage of the M-measurement-optimal feedback control over other suboptimal control algorithms is that one or more future observations are incorporated into the control computations. This allows for the "probing" or "test signal" type of control action which is characteristic of the truly optimal stochastic control [2]. The open-loop-optimal feedback control cannot take this form since future measurements are ignored.

Extensions The concept of incorporating M future measurements rather than the actual number of measurements may be used to simplify computations involving constrained controls, nonlinear plants, and nonquadratic cost criteria as well. For example, if the actual controls are constrained at all future times, it can be assumed for the purposes of control computation that they are constrained only at M future times and unconstrained at all other times. Similarly, it can be assumed that the plant is nonlinear only at M future times but linear at all other times, or that the cost is nonquadratic at M future times, but quadratic at all other times.

In all of these situations the suboptimal control approaches the optimal control as M approaches the number of stages remaining in the problem. Generally speaking, the computational effort will increase as M increases, but the author knows of no analysis to show that the performance will improve as M increases.

III. ONE-MEASUREMENT-OPTIMAL FEEDBACK CONTROL

Problem Statement

Given:

- a) The state equation

$$x_{i+1} = f_i(x_i, u_i, w_i) \quad i = 0, 1, \dots, N \quad (1)$$

where x_i is the state vector at time t_i , u_i is the control vector at time t_i , and w_i is the process noise at time t_i .

b) The measurement equation

$$z_i = h_i(x_i, v_i) \quad i = 1, 2, \dots, N \quad (2)$$

where v_i is the observation noise at time t_i .

c) The cost criterion

$$J = \mathcal{E} \left\{ \sum_{i=0}^N L_i(x_i, u_i) + \phi(x_{N+1}) \right\} \quad (3)$$

d) The probability density functions $p(x_0)$, $p(w_i)$, $p(v_i)$. It is assumed that the vectors $x_0, \{w_i\}$, and $\{v_i\}$ are independent.

e) For notational convenience let

$$Z_k = \{z_1, z_2, \dots, z_k\}$$

$$U_k = \{u_0, u_1, \dots, u_k\}$$

Open-Loop-Optimal Feedback Control

The cost function to be minimized by the open-loop control at time t_k is given by

$$J_{OL,k} = \mathcal{E} \left[\sum_{j=k}^N L_j(x_j, u_j) + \phi(x_{N+1}) \mid Z_k, U_{k-1} \right] \quad (4)$$

This is minimized by the control sequence $\{u_k, \dots, u_N\}$ subject to the state constraint (1). The first member of this sequence, u_k , is the open-loop-optimal feedback control at time t_k .

One-Measurement-Optimal Feedback Control

This portion treats a special case of M-measurement-optimal feedback control. If only one future measurement is incorporated in the control computation at time t_k , and if this measurement occurs at time t_{k+n} , then the one-measurement cost function becomes

$$J_{OM,k} = \mathcal{E} \left\{ \sum_{j=k}^{k+n-1} L_j(x_j, u_j) + \mathcal{E} [J_{OL,k+n}^0 \mid Z_k, U_{k+n-1}] \mid Z_k, U_{k-1} \right\} \quad (5)$$

In this equation, the optimum open-loop cost function $J_{OL,k+n}^0$ is similar to (4) after minimization, but it depends on the measurements $\{z_k, z_{k+n}\}$ and the controls U_{k+n-1} . The first member of the control sequence $\{u_k, \dots, u_N\}$ that minimizes (5) is the one-measurement-optimal feedback control at time t_k .

The solution to (5) may still be difficult to find in practice. Even if the explicit dependence of the open-loop cost function on the open-loop-optimal controls $\{u_{k+n}, \dots, u_N\}$ can be removed, the solution of (5) still involves an n-stage two point boundary problem. If, however, the future

measurement occurs at the "next" time, then (5) reduces to a parameter optimization over u_k , i.e., the one-measurement cost function becomes

$$J_{OM,k} = \mathcal{E} \left\{ L_k(x_k, u_k) + \mathcal{E} \left[J_{OL,k+1}^0 \mid Z_k, U_k \right] \mid Z_k, U_{k-1} \right\} \quad (6)$$

In this equation $J_{OL,k+1}^0$ depends on the measurements Z_{k+1} and the controls U_k .

Linear Systems, Quadratic Cost, and Nonlinear Measurements

This section contains some specialized results when the state equation is linear

$$x_{i+1} = \Phi_i x_i + G_i u_i + w_i \quad i = 0, \dots, N \quad (7)$$

and when the cost is quadratic

$$J = \mathcal{E} \left\{ \sum_{i=0}^N x_i^T A_i x_i + u_i^T B_i u_i + x_{N+1}^T A_{N+1} x_{N+1} \right\} \quad (8)$$

Here the superscript T denotes the transpose, and the weighting matrices $\{A_i\}$, $\{B_i\}$ are positive definite. The measurements may still be nonlinear.

It has been shown [3, Appendix D] that the optimum open-loop cost function in this case is given by

$$J_{OL,k+1}^0 = \mathcal{E}(x_{k+1}^T S_{k+1} x_{k+1} \mid Z_{k+1}, U_k) + \text{tr} [E_{k+1}(F_{k+1} - S_{k+1})] + \text{const.} \quad (9)$$

where tr is the trace of a matrix, and

$$S_i = A_i + \Phi_i^T [S_{i+1} - S_{i+1} G_i (B_i + G_i^T S_{i+1} G_i)^{-1} G_i^T S_{i+1}] \Phi_i \quad (10)$$

$$S_{N+1} = A_{N+1}$$

$$F_i = \Phi_i^T F_{i+1} \Phi_i + A_i \quad (11)$$

$$F_{N+1} = A_{N+1}$$

$$E_{k+1} = \text{covariance of } x_{k+1} \text{ conditioned on } Z_{k+1} \text{ and } U_k.$$

A by-product in the derivation of (9) is the open-loop-optimal feedback control. At time t_k it is given by [3, p.80]

$$u_{OL,k}^0 = - (B_k + G_k^T S_{k+1} G_k)^{-1} G_k^T S_{k+1} \Phi_k \mathcal{E}(x_k \mid Z_k, U_{k-1}) \quad (12)$$

This is the precise form of the truly optimal control when the measurements are linear, the plant is linear, and the cost is quadratic.

The equation for the one-measurement cost function with the future measurement occurring at the next time, Eq.(6), becomes

$$J_{OM,k} = \mathcal{E} \left[x_k^T A_k x_k + u_k^T B_k u_k + x_{k+1}^T S_{k+1} x_{k+1} \right. \\ \left. + \mathcal{E} \left\{ \text{tr} \left[E_{k+1} (F_{k+1} - S_{k+1}) \right] \middle| Z_k, U_k \right\} \middle| Z_k, U_{k-1} \right] \quad (13) \\ + \text{const.}$$

The value of u_k that minimizes (13) is the one-measurement-optimal feedback control at time t_k for linear systems, quadratic cost, and nonlinear measurements. This is computationally equivalent to solving for the first control action in a two stage process [3, p.67]. Nonlinear control action results from the fact that the conditional covariance E_{k+1} in (13) is a function of u_k . (See [3] for some numerical results of the two stage control process.) It has also been shown in [3, Appendix D] that the weighting matrix $F_{k+1} - S_{k+1}$ is at least positive semidefinite. If it were not, the control would minimize the cost by degrading, not improving, the knowledge of the state.

IV. SIMULATIONS

System Configuration

For definiteness, let the scalar state satisfy the equation

$$x_{i+1} = x_i + u_i + w_i \quad i = 0, \dots, N \quad (14)$$

where x is the state, u is the scalar control, and w is Gaussian process noise. The initial conditions also have a Gaussian distribution. The cost function is quadratic in nature.

$$J = \mathcal{E} \left\{ \sum_{i=0}^N A_i x_i^2 + B_i u_i^2 + A_{N+1} x_{N+1}^2 \right\} \quad (15)$$

Observations are taken through a three level quantizer shown in Fig. 1. Observation noise is not included in this example.

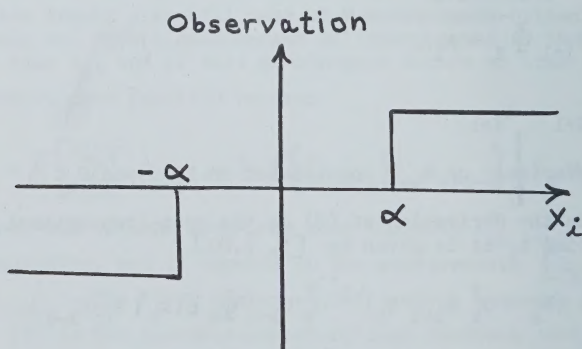


Fig. 1. Three Level Quantizer for Observations

Estimation Procedure

Obtaining numerical solutions to the stochastic control problem requires extensive use of the probability density function of the state conditioned on past measurements and control actions. Here it is assumed that the distribution just prior to a quantized measurement is Gaussian. This algorithm has been extensively treated in [3], and appears to be a good approximation for linear systems and quantized measurements. The details of calculating moments conditioned on quantized measurements can also be found in [5].

Simulation Description

Both the open-loop-optimal feedback control and the one-measurement-optimal feedback control have been simulated on the digital computer. The latter algorithm assumes that the one future measurement occurs at the next time instant. The control algorithms used the same realizations of the initial conditions and process noise, although the state trajectories, in general, were different. Each case was run 50 times and the ensemble averages were approximated by the numerical average of these 50 trials. In all cases the weighting coefficients for the control effort $\{B_i\}$ are unity, and the process noise variance is constant, but varies from case to case.

Numerical Results

The results of a seven stage stochastic process are described. Quantitative results are displayed for a terminal control problem, and the results of other simulations are discussed.

Terminal Control In the terminal control problem, all state weighting coefficients are unity except the last one, A_7 , which is 100. This means, roughly, that the RMS terminal error is ten times "more important" than the other quantities in the cost function.

Fig. 2 shows the results when the measurements are taken through a three level quantizer: the ensemble mean square state and ensemble average of the (approximate) conditional covariance of the state are plotted as a function of time. For this case, the variance of the process noise is 0.2, and the quantizer switch points are at ± 1 .

The most noticeable difference between the two control laws is that the one-measurement control acts to reduce the conditional covariance of the state estimate. Note that the ensemble average of the conditional covariance is about half the average of the conditional covariance for the open-loop control. The one-measurement control is able to do this by centering the conditional distribution of the measurement near the quantizer switch point. This is reflected in the curves for the mean square value of the state, which stays in the neighborhood of 1.0 (the switch point) for the one-measurement control, but gradually goes to zero for the open-loop control. The control effort (not shown) for the one-measurement control is higher, and it requires a large control action at the last application to bring the state from the vicinity of the quantizer switch point to the origin.

The performance penalty of the open-loop-optimal feedback control over the one-measurement-optimal feedback control is 17.1 percent for this case. Other simulations revealed that the performance penalty ranged as high as 44 percent when observations were taken through a two level quantizer.

Other Simulations Cost functions other than the terminal control type were simulated: the state deviations were weighted more heavily as time progressed, or else the weightings were constant. The performance advantage

of the one-measurement control was always less than 10 percent in these cases. This arises from the fact that the one-measurement control tries to move the state around to gain information, but these movements are restricted by the heavy weighting on the state deviations.

Thus a qualitative assessment, at least for linear systems and nonlinear measurements, is that incorporating future measurements in the control computations will yield the greatest return when the cost function is such that the state and/or control is free to reduce uncertainty in the estimate. In other situations, the open-loop control is quite attractive, especially because of its computational simplicity.

V. CONCLUSIONS

A new suboptimal stochastic control algorithm is presented which incorporates one or more future measurements into the control computations. The two limiting forms of the algorithm are the well-known open-loop-optimal feedback control and the truly optimal stochastic control. This concept can be extended to simplify the computations for constrained controls, nonlinear plants, and nonquadratic cost criteria. A linear system with quantized measurements was simulated to compare the one-measurement-optimal feedback control to the open-loop-optimal feedback control. The greatest improvement over the open-loop algorithm occurs in those situations where the cost function gives the control and state some freedom to reduce the uncertainty in the state estimate.

ACKNOWLEDGEMENT

The author wishes to acknowledge the helpful discussions of C. F. Price of the Experimental Astronomy Laboratory, MIT.

REFERENCES

- [1] Dreyfus, S., "Some Types of Optimal Control of Stochastic Systems", J. SIAM Control, Ser. A, Vol.2, pp.120-134.
- [2] Fel'dbaum, A., "Dual Control Theory, I-IV", Automation and Remote Control, 21, 1960, p.874, p.1033; 22, 1961, p.1, p.109.
- [3] Curry, R. E., "Estimation and Control with Quantized Measurements", Ph.D. Dissertation, Dept. Aeronautics and Astronautics, MIT, Cambridge, Mass., May, 1968. Also Report TE-23, Experimental Astronomy Laboratory, MIT, Cambridge, Mass.

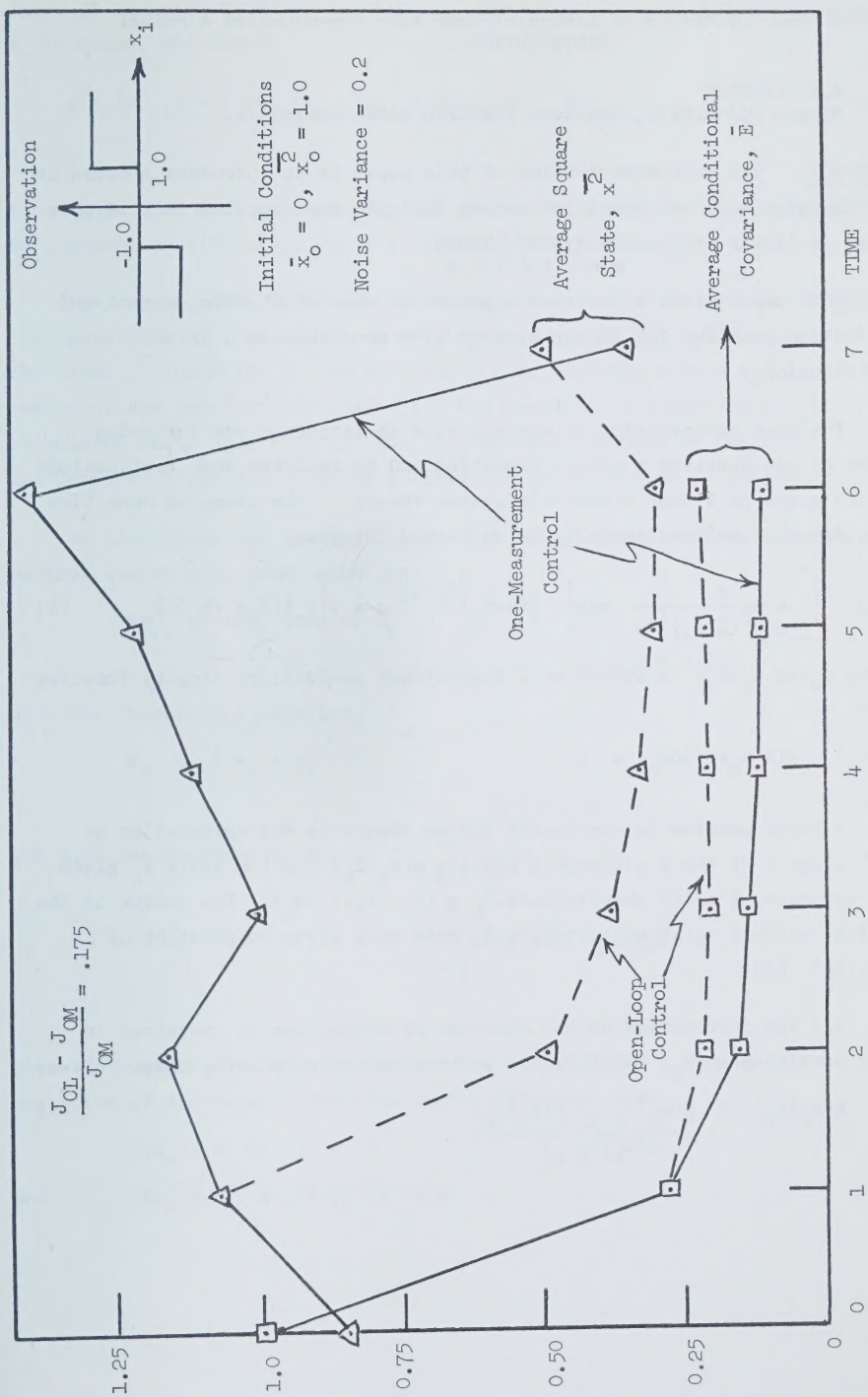


Fig. 2. Seven Stage Terminal Stochastic Control with Observations Quantized to Three Levels

CONTROL AND ESTIMATION OF LINEAR SYSTEMS WITH NON-GAUSSIAN A PRIORI DISTRIBUTIONS.

A.V. CAMERON

Monash University, Clayton, Victoria 3168, Australia.

ABSTRACT. The main contribution of this paper is to introduce a class of non-Gaussian a priori densities and to indicate some applications of this class in linear stochastic system theory.

* * * * *

This paper gives a Bayesian approach to some stochastic control and estimation problems for linear systems with non-Gaussian a priori distributions.

The main contribution of the paper is to introduce the following class of non-Gaussian a priori densities and to indicate some applications of the class in linear stochastic system theory. The class of densities considered is written formally as an n-fold integral,

$$p(x) = \int_{-\infty}^{\infty} \frac{1}{(2\pi)^{n/2} |P_0|^{1/2}} \exp \left[-\frac{1}{2} (x - \hat{x}_0)^T P_0^{-1} (x - \hat{x}_0) \right] c_0 f(A_0 \hat{x}_0 + b_0) d\hat{x}_0 \quad (1)$$

where $c_0 f(A_0 \hat{x}_0 + b_0)$ is itself an n-dimensional probability density function with

$$\int_{-\infty}^{\infty} c_0 f(A_0 \hat{x}_0 + b_0) d\hat{x}_0 = 1$$

A basic problem in stochastic system theory is the calculation at each stage k of the a posteriori density $p(x_k | Z_k)$ for the state x_k given the sequence of noisy observations $Z_k = (z_1, z_2, \dots, z_k)$. The choice of the optimal control u_k at each stage k is made only after computation of $p(x_k | Z_k)$ [1].

All the information of the observer or controller is contained in this density $p(x_k | Z_k)$ which can be written recursively using Bayes rule as

$$p(x_k | Z_k) = \frac{p(x_k | Z_{k-1}) p(z_k | x_k)}{p(z_k | Z_{k-1})} \quad (2)$$

$$k = 1, 2, 3, \dots$$

where it is assumed the observations z_k are conditionally independent when x_k is known, and where

$$p(x_k | z_{k-1}) = \int p(x_k | x_{k-1}, u_{k-1}) p(x_{k-1} | z_{k-1}) dx_{k-1} \quad (3)$$

$$k = 1, 2, 3, \dots$$

the sequence of densities $p(x_k | z_k)$ is started by assuming in equation (3) an a priori density $p(x_0)$ for the initial state vector x_0 .

The key property of the non-Gaussian densities (1) is that under both the operations (2) and (3) where $p(z_k | x_k)$ and $p(x_k | x_{k-1}, u_k)$ are Gaussian, the class of densities is "reproducing", i.e. starting with $p(x_0)$ as a member of the non-Gaussian class (1), the sequence of densities $p(x_k | z_k)$, calculated using equations (2) and (3), are each members of the same non-Gaussian class described by a fixed and finite number of parameters.

To illustrate the application of this result, consider the scalar control system with plant equation

$$x_{k+1} = Fx_k + Gu_k + \omega_k \quad (4)$$

$$-\infty < u_k < \infty, \quad 0 \leq k \leq N-1$$

with the observation equation,

$$z_k = Hx_k + v_k \quad (5)$$

$$0 \leq k \leq N-1$$

and with the performance index $J = \sum_{k=1}^n W_k(x_k, u_{k-1})$

$$= \sum_{k=1}^N (s_k x_k^2 + t_{k-1} u_{k-1}^2) \quad (6)$$

$$t_{k-1}, s_k > 0$$

where F , G and H are known parameters and where ω_k and v_k are each sequences of independent Gaussian random variables with

$$E(\omega_k) = E(v_k) = 0$$

and $E(\omega_k^2) = Q, E(v_k^2) = R.$

Assume also that x_0 has a non-Gaussian a priori density of the form of equation (1),

$$\text{i.e. } p(x_0) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi P_0}} \exp\left[-\frac{1}{2P_0} (x_0 - \hat{x}_0)^2\right] c_0 f(A_0 \hat{x}_0 + b_0) d\hat{x}_0 \quad (7)$$

where $c_0 f(A_0 \hat{x}_0 + b_0)$ is a known probability density function.

The form of the a priori distribution $p(x_0)$ is particularly important in optimal control problems which involve only a finite number of stages N as indicated in equation (6), and the extension of the class of a priori distributions from the usual Gaussian class is important in practice since the a priori distribution $p(x_0)$ representing all the initial objective and/or subjective information of the observer about x_0 can have a large variety of shapes and forms, e.g. multimodal and skew.

Starting with the a priori density (7), the sequence of densities $p(x_k | Z_k)$ can be calculated recursively using the two equations (2) and (3), and

$$p(x_k | Z_k) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi P_k}} \exp\left[-\frac{1}{2P_k} (x_k - \hat{x}_k)^2\right] c_{k-1} f(A_{k-1} \hat{x}_{k-1} + b_{k-1}) d\hat{x}_{k-1} \quad (8)$$

$$k = 1, 2, 3, \dots, N-1$$

where

$$\hat{x}_k = F \hat{x}_{k-1} + G u_{k-1} + K_k \left[z_k - H(F\hat{x}_{k-1} + Gu_{k-1}) \right] \quad (9)$$

where

$$K_k = \frac{H_k (Q + F^2 P_{k-1})}{H^2 (Q + F^2 P_{k-1}) + R} \quad (10)$$

and where

$$\frac{1}{P_k} = \frac{H^2}{R^2} + \frac{1}{(Q + F^2 P_{k-1})} \quad (11)$$

Equation (9), (10) and (11) are the usual Kalman-Bucy relations for the linear Gaussian system [1,2].

Changing the variable in the integral (8) from $\hat{x}_{k-1}$ to $\hat{x}_k$ using the linear transformation equation (9),

$$p(x_k | Z_k) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi P_k}} \exp\left[-\frac{1}{2P_k} (x_k - \hat{x}_k)^2\right] c_k f(A_k \hat{x}_k + b_k) d\hat{x}_k \quad (12)$$

where

$$\begin{aligned} A_k &= A_{k-1}(F - K_k HF)^{-1} \\ b_k &= b_{k-1} - A_k \left[Gu_{k-1} + K_k(z_k - HGu_{k-1}) \right] \\ c_k &= c_{k-1}(F - K_k HF)^{-1} \end{aligned} \quad (13)$$

Thus starting from the a priori density $p(x_0)$, equation (7), the sequence of densities $p(x_k|z_k)$ $k = 0, 2, 3, \dots, N-1$ each belongs to the same family of densities, equation (12), described at each stage k by the parameters P_k, A_k, b_k, c_k .

The set of recursive equations (13), (10) and (11) thus completely specify the density $p(x_k|z_k)$ at each stage k and a complete solution to the optimal estimation problem can now be obtained once an error criterion is specified.

The optimal control policy can in principle be obtained by the usual dynamic programming procedure starting from the boundary condition,

$$V_{N-1}(z_{N-1}) = \min_{u_{N-1}} E_{x_N|z_{N-1}} [W_N(x_N, u_{N-1})] \quad (14)$$

and using the recurrence relation,

$$V_{k-1}(z_{k-1}) = \min_{u_{k-1}} \left[E_{x_k|z_{k-1}} (W_k(x_k, u_{k-1})) + E_{z_k|z_{k-1}} (V_k(z_k)) \right] \quad (15)$$

At the last stage, where the optimal control u_{N-1}^* has to be determined,

$$E_{x_N|z_{N-1}} (W_N(x_N, u_{N-1})) = E_{x_N|z_{N-1}} (s_N x_N^2 + t_{N-1} u_{N-1}^2)$$

where the expectation is taken over the density $p(x_N|z_{N-1})$ given by equation (3).

Thus

$$\begin{aligned} E_{x_N|z_{N-1}} W_N(x_N, u_{N-1}) &= E_{x_{N-1}|z_{N-1}} \left[s_N (Fx_{N-1} + Gu_{N-1})^2 + s_N Q + t_{N-1} u_{N-1}^2 \right] \\ &= s_N (Q + F^2 R_{N-1}) + t_{N-1} u_{N-1}^2 + s_N (Fu_{N-1} + Gu_{N-1})^2 \end{aligned} \quad (16)$$

$$\text{where } \mu_{N-1} = E(x_{N-1} | Z_{N-1})$$

$$R_{N-1} = E((x_{N-1} - \mu_{N-1})^2 | Z_{N-1})$$

Minimising equation (16) with respect to μ_{N-1} , the optimal control u_{N-1}^* is given by

$$u_{N-1}^* = -\Delta_{N-1} \mu_{N-1}$$

$$\text{where } \Delta_{N-1} = \frac{s_N G F}{t_{N-1} + s_N G^2}$$

But equation (16) has the same form in terms of μ_{N-1} and R_{N-1} regardless of the particular non-Gaussian density $p(x_{N-1} | Z_{N-1})$. Hence for the quadratic performance index, only the means and variances of $p(x_{N-1} | Z_{N-1})$ need be calculated and the non-Gaussian densities $p(x_{N-1} | Z_{N-1})$ are unnecessary. The complete control policy is identical to the policy for Gaussian a priori densities [2].

For a performance index other than quadratic, the calculation of $E_{x_N | Z_{N-1}}(w(x_N, u_{N-1}))$ and its minimisation with respect to u_{N-1} , are difficult nonlinear problems and the optimal control can only be found numerically.

However using the reproducing property of the densities $p(x_k | Z_k)$, equations (13), (11) and (10), the one stage optimisation, where each control stage is optimised separately, gives a fixed nonlinear control function $u_k = u_k(z_k)$ and this is given by

$$\min_{u_{k-1}} \int w(x_k, u_{k-1}) p(x_k | Z_{k-1}) dx_k$$

where $p(x_k | Z_{k-1})$ is given by equation (3) and is of fixed functional form for each k , $k = 1, 2, \dots, N-1$.

1. Ho, Y.C. and Lee, R.C.K. "A Bayesian Approach to Problems in Stochastic Estimation and Control." IEEE Trans. on Automatic Control, Vol. AC-9, pp.333-339, October 1964.
2. Aoki, M. "Optimisation of Stochastic Systems." Academic Press N.Y., London, 1967.

D. ISAACS

Associate Professor, University of California, Irvine - Irvine, Calif.

W. D. LOCHRIE

North American Rockwell Fellow, University of California, Irvine
Irvine, California

ABSTRACT

A recursive estimation algorithm is developed for a system with linear dynamics but in which constraints are imposed on the state variables. The estimates are chosen to minimize the squared observation error subject to constraints which are convex in the state variables. In problems in which this constraint set does not have to be approximated the estimator is optimal. Cases in which the convex constraint set must be approximated result in a suboptimal filter, which in the limit of imposing no constraints, reduce to the Kalman optimal linear filter. A nonlinear programming computational algorithm is used in the estimation method developed.

INTRODUCTION

The quality of the estimate of the state of a system is obviously improved as more information concerning the system is used. For example, the Kalman filter for linear systems [1] which uses the estimation error covariance matrix and the measurement noise covariance provides better estimates than a least squares filter which only uses the measurement noise statistics. Similarly, if physical or mathematical reasoning imposes constraints on the state variables, the estimate of the states will be improved if these constraints are implemented in the estimation algorithm.

There are, of course, many ways to approach this problem of estimation with constraints. The estimate obtained by simply using linear filtering methods and then "readjusting" the estimates which exceed the constraints is certainly the simplest suboptimal method to implement. At the other end of the spectrum, implementation of the optimal algorithm is impractical for many problems. This paper gives a practical computational method for solving this problem.

As pointed out by Ho [2], estimation problems can be approached as a problem of minimizing a quadratic form. It is in this context that the estimation problem with constrained state variables is formulated. This formulation makes the problem amenable to solution using nonlinear programming (NP) techniques.

One advantage of this method of solution stems from the fact that there exists an array of NP techniques that can be used to solve constrained minimization problems quite rapidly on currently available computers. Also, recursive relations for the quadratic objective function are developed in Section III making implementation of the algorithm practical for estimation problems. The particular NP algorithm chosen depends on the specific problem. For example, if the constraints are linear, Wolfe's method [3] for solving quadratic programming problems can be used. Either Rosen's method [4] or the Lagrangian differential gradient technique [5] can be used when the constraints are nonlinear. Hadley [6] discusses these and other methods in detail.

II. PROBLEM FORMULATION

Consider a system described by the dynamical equation:

$$\dot{\underline{x}}(t) = F(t) \underline{x}(t), \quad t \in [0, T] \quad (1)$$

where $\underline{x}(t)$ is an n element vector called the state vector. The solution to this differential equation at time t_{j+1} is given by

$$\underline{x}(t_{j+1}) = \phi(t_{j+1}, t_j) \underline{x}(t_j) \quad (2)$$

where $\phi(t_{j+1}, t_j)$ is the transition matrix. It is obtained by integrating from t_j to t_{j+1} the matrix differential equation

$$\dot{\phi}(t, t_j) = F(t) \phi(t, t_j) \quad \text{with } \phi(t_j, t_j) = I \quad (3)$$

For notational convenience let ϕ_j denote $\phi(t_{j+1}, t_j)$ and let the subscript j on the other matrices and vectors denote evaluation at time t_j . Assume that the state variables are constrained such that

$$\underline{g}(\underline{x}(t)) \leq \underline{b} \quad t \in [0, T] \quad (4)$$

where $\underline{g}$, a vector of convex functions of $\underline{x}(t)$, defines a closed convex set C in E_n . Let the observations be made at $t_1, t_2, \dots, t_k \dots$ and assume they are related to the states by

$$\underline{y}_k = M_k \underline{x}_k + \underline{v}_k \quad (5)$$

The m element vector $\underline{v}$ is zero mean white measurement noise.

Thus it is desired to obtain an algorithm which uses the observation $\underline{y}_k$ to update the state estimate $\hat{\underline{x}}_k$ such that the squared observation errors are minimized subject to constraint (4).

III. DERIVATION OF ALGORITHM

It is desirable to update the estimate of $\underline{x}_{k+1}$ in a recursive fashion. That is, given an estimate $\hat{\underline{x}}_k$ based on $\underline{y}_1, \dots, \underline{y}_k$, compute a new estimate $\hat{\underline{x}}_{k+1}$ based just on $\underline{y}_{k+1}$ and $\hat{\underline{x}}_k$. To this end first consider the problem of "batch processing" the data. That is, determine $\hat{\underline{x}}_{k+1}$ based on all the data $\underline{y}_1, \dots, \underline{y}_{k+1}$.

The expected measurement at t_i is given by

$$\hat{\underline{y}}_i = M_i \phi(t_i, t_{k+1}) \hat{\underline{x}}_{k+1} \quad (6)$$

For notational convenience make the following definitions

$$M_k^* \triangleq \begin{bmatrix} M_1 \phi(t_1, t_k) \\ \vdots \\ M_k \phi(t_k, t_k) \end{bmatrix}, \quad \hat{\underline{y}}_k^* \triangleq \begin{bmatrix} \hat{\underline{y}}_1 \\ \vdots \\ \hat{\underline{y}}_k \end{bmatrix} \quad \text{and} \quad \underline{y}_k^* \triangleq \begin{bmatrix} \underline{y}_1 \\ \vdots \\ \underline{y}_k \end{bmatrix} \quad (7)$$

where the $\underline{y}_i$ denote the vectors of actual measurements.

Considering all the measurements yields

$$\begin{bmatrix} M_k^* \phi_k^{-1} \\ M_{k+1} \end{bmatrix} \hat{x}_{k+1} = \begin{bmatrix} \hat{y}_k^* \\ \hat{y}_{k+1} \end{bmatrix} \quad (8)$$

The measurements are not exact because of noise, thus it is desirable to determine $\hat{x}_{k+1}$ such that the following functional is minimized,

$$J_{k+1} = \left\| \begin{bmatrix} M_k^* \phi_k^{-1} \\ M_{k+1} \end{bmatrix} \hat{x}_{k+1} - \begin{bmatrix} y_k^* \\ y_{k+1} \end{bmatrix} \right\|_{Q_{k+1}^*}^2 \quad (9)$$

subject to the constraint $\underline{g}(\phi(t, t_{k+1}) \hat{x}_{k+1}) \leq \underline{b}$, $t \in [0, T]$.

The notation $\|s\|_{Q_{k+1}^*}$ denotes the quadratic form $\underline{s}^T Q_{k+1}^* \underline{s}$ where Q_{k+1}^* is a positive definite matrix.

Assume that Q_{k+1}^* is of the form

$$Q_{k+1}^* = \begin{bmatrix} Q_k^* & 0 \\ 0 & Q_{k+1} \end{bmatrix} \quad (10)$$

Then, expanding the quadratic form in Equation (9) leads to the relation

$$J_{k+1} = \left\| M_k^* \phi_k^{-1} \hat{x}_{k+1} - y_k^* \right\|_{Q_k^*}^2 + \left\| M_{k+1} \hat{x}_{k+1} - y_{k+1} \right\|_{Q_{k+1}}^2 \quad (11)$$

Expanding J_{k+1} , eliminating terms not containing $\hat{x}_{k+1}$ and denoting the remaining terms by the functional l_{k+1} yields

$$\begin{aligned} l_{k+1} = & \hat{x}_{k+1}^T [(\phi_k^{-1})^T M_k^{*T} Q_k^* M_k^* \phi_k^{-1} + M_{k+1}^T Q_{k+1} M_{k+1}] \hat{x}_{k+1} \\ & - 2[y_k^{*T} Q_k^* M_k^* \phi_k^{-1} + y_{k+1}^T Q_{k+1} M_{k+1}] \hat{x}_{k+1} \end{aligned} \quad (12)$$

It is desired to get a recursive relation for the terms in l_{k+1} . Careful observation of Equation 12 and noting that the starred matrices are augmented makes the following definitions apparent.

$$D_k \triangleq M_k^{*T} Q_k^* M_k^* \quad (13)$$

and

$$D_{k+1} \triangleq (\phi_k^{-1})^T D_k \phi_k^{-1} + M_{k+1}^T Q_{k+1} M_{k+1} \quad (14)$$

$$c_k^T \triangleq 2y_k^{*T} Q_k^* M_k^* \quad (15)$$

and

$$c_{k+1}^T \triangleq c_k^T \phi_k^{-1} + 2y_{k+1}^T Q_{k+1} M_{k+1} \quad (16)$$

The problem is converted to a maximization problem by changing the sign of l_{k+1} and denoting it by $f(\hat{x}_{k+1})$. Thus using the above recursive relations and Equation 12 the problem can now be stated as a NP problem:

$$\text{Maximize } f(\hat{x}_{k+1}) = c^T \hat{x}_{k+1} - \hat{x}_{k+1}^T D_{k+1} \hat{x}_{k+1} \quad (17)$$

$$\text{Subject to } g(\phi(t, t_{k+1}) \hat{x}_{k+1}) \leq b, \quad t \in [0, T]$$

In order to obtain a numerical solution it is necessary to approximate the constraints. This involves eliminating constraints that are never binding and either using the remaining effective constraints or approximating the region with convex functions. In other words, it is necessary to obtain a set of convex functions h_i such that the constraints

$$h(\hat{x}_{k+1}) \leq d \quad (18)$$

approximate the constraints in (17) as closely as the specific problem requires. In order to solidify this idea an example with two state variables is given. Consider an LC circuit in which the state variables are the charge on the capacitor, $q(t)$, and the current through the inductor, $i(t)$. The transition matrix is given by the following.

$$\phi(t, t_{k+1}) = \begin{bmatrix} \cos \omega(t - t_{k+1}) & \frac{-1}{\omega} \sin \omega(t - t_{k+1}) \\ \omega \sin \omega(t - t_{k+1}) & \cos \omega(t - t_{k+1}) \end{bmatrix} \quad (19)$$

where $\omega = \frac{1}{\sqrt{LC}}$. Assume for convenience that $\omega = 1$ and that the

constraints are upper bound type constraints. Denote the maximum values of $q(t)$ and $i(t)$ by q_m and i_m respectively and assume that $q_m < i_m$.

Thus for this example the constraints in (17) are linear and can be written as shown in Figure 1.

These constraints are drawn in Figure 1 for various values of time t . It is apparent from the figure that for this example equation (18) is simply one constraint,

$$\hat{i}_{k+1}^2 + \hat{q}_{k+1}^2 \leq q_m^2.$$

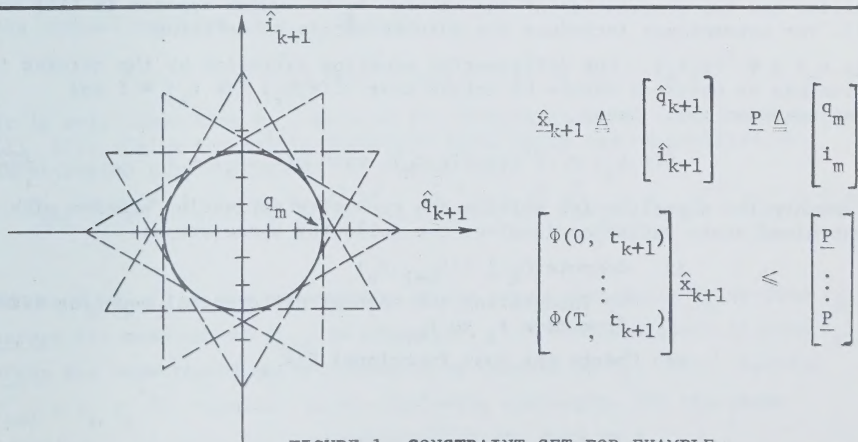


FIGURE 1 CONSTRAINT SET FOR EXAMPLE

Returning to the general problem, note that when a new data vector y_{k+1} is being processed the estimate $\hat{x}_{k+1}$ is a feasible solution and can be used as an initial solution in the NP algorithm.

The statistical significance of the matrices O_k and D_k will be mentioned now and explained in detail in section VI. The matrix O_k is the inverse of the measurement noise covariance matrix. That is since $E[y_k] = 0$,

$$O_k = [E\{v_k v_k^T\}]^{-1} \Delta R_k^{-1} \quad (20)$$

The initial estimate $\hat{x}_0$ is assumed to be chosen such that inequality (4) holds strictly. Then the matrix D_0 is the inverse of the estimation error covariance matrix at t_0 . That is, assuming that $g(\hat{x}_0) < \underline{b}$,

$$D_0^{-1} = E\{(\hat{x}_0 - x_0)(\hat{x}_0 - x_0)^T\} \Delta P_0 \quad (21)$$

Subsequent values of D_k are not necessarily related to the estimation error covariance. However, at each stage the estimate minimizes the norm (9) subject to constraint (18).

Also note that regardless of which NP technique is used, convergence to an estimate which is a global optimum is guaranteed at each stage. This can be proven by showing that $f(\hat{x}_k)$ is a strictly concave function for each k . First note that D_0 and O_k , $k=1,2,\dots$, are positive definite since D_0^{-1} and O_k are covariance matrices and hence are positive definite. Since Φ_k^{-1} is a nonsingular transformation $(\Phi_k^{-1})^T D_0 \Phi_k^{-1}$ is also positive definite. The symmetry and positive definiteness of $\Phi_k^T O_k \Phi_k$ imply that there exists a nonsingular matrix P_k such that $O_k = P_k P_k^T$. Therefore the matrix $M_k O_k M_k^T$ is positive semidefinite for each k . Hence, D_1 is positive definite. Repetition of the above argument shows that at each stage D_k is positive definite. Since $-\hat{x}_{k+1}^T D_{k+1} \hat{x}_{k+1}$ is a negative definite quadratic form, it is strictly concave. Thus $f(\hat{x}_{k+1})$, given in (17) is a strictly concave function over a closed set C in E_{k+1} . This means that any relative maximum of $f(\hat{x}_{k+1})$ in C is also a global maximum of $f(\hat{x}_{k+1})$ in C [6].

Consider now the computation of the matrix Φ_k^{-1} which appears in (14) and (16). For convenience introduce the adjoint matrix by defining

$\psi^T(t, t_k) \Delta \Phi^{-1}(t, t_k)$. The differential equation satisfied by the adjoint system can be obtained easily by noting that $\psi^T(t, t_k) \Phi(t, t_k) = I$ and using equation (3). Hence,

$$\dot{\psi}^T(t, t_k) = -F^T(t) \psi^T(t, t_k) \quad \text{with } \psi^T(t_k, t_k) = I \quad (22)$$

In summary the algorithm for solving the recursive estimation problem with constrained state variables involves the following three steps:

- i) Compute $\psi_k \Delta \psi(t_{k+1}, t_k)$
by integrating the adjoint differential equation (22) from time t_k to t_{k+1} .
- ii) Update the cost functional $f(\hat{x}_{k+1})$

$$\begin{aligned}
1. \quad D_{k+1} &= \psi_k D_k \psi_k^T + M_{k+1}^T Q_{k+1} M_{k+1} \text{ with } D_0 = [\text{cov}(\hat{x}_0)]^{-1} \\
2. \quad c_{k+1} &= \psi_k c_k + 2 M_{k+1}^T Q_{k+1} y_{k+1} \text{ with } c_0 = 2 D_0 \hat{x}_0 \\
3. \quad f(\hat{x}_{k+1}) &= \hat{x}_{k+1}^T (-D_{k+1}) \hat{x}_{k+1} + c_{k+1} \hat{x}_{k+1}
\end{aligned} \tag{23}$$

iii) Solve the NP problem

Maximize $f(\hat{x}_{k+1})$ subject to $h(\hat{x}_{k+1}) \leq d$

where the initial estimate is assumed to be $\phi(t_{k+1}, t_k) \hat{x}_k$.

IV. UNCONSTRAINED STATE VARIABLES AND THE KALMAN FILTER

Consider now the case when the constraints are not activated either because the approximation to the constraint set is too "loose" or possibly the signal/noise ratio is high. Since the state variables are not constrained, it is possible to differentiate the cost functional with respect to the state variables and set the resulting expression equal to zero. Thus, at time t_{k+1}

$$D_{k+1} \hat{x}_{k+1} = \frac{1}{2} c_{k+1} \tag{24}$$

Substituting Equation (16) into (24) yields the expression

$$D_{k+1} \hat{x}_{k+1} = M_{k+1}^T Q_{k+1} y_{k+1} + (\phi_k^{-1})^T (\frac{1}{2} c_k) \tag{25}$$

Since (24) also holds at t_k , Equation (25) becomes

$$D_{k+1} \hat{x}_{k+1} = M_{k+1}^T Q_{k+1} y_{k+1} + (\phi_k^{-1})^T D_k \hat{x}_k \tag{26}$$

Postmultiplying Equation (14) by $\phi_k \hat{x}_k$ and using (26) yields

$$D_{k+1} \hat{x}_{k+1} = M_{k+1}^T Q_{k+1} [y_{k+1} - M_{k+1} \phi_k \hat{x}_k] + D_{k+1} \phi_k \hat{x}_k \tag{27}$$

Thus using the definition given in Equation (20) and letting $D_k = P_k^{-1}$ the optimal estimate at t_{k+1} becomes

$$\hat{x}_{k+1} = P_{k+1} M_{k+1}^T R_{k+1}^{-1} [y_{k+1} - M_{k+1} \phi_k \hat{x}_k] + \phi_k \hat{x}_k \tag{28}$$

It is well known that P_{k+1} denotes the estimation error covariance matrix [1], [2]. Using the matrix inversion lemma which can be verified by substitution permits (14) to be written as

$$P_{k+1} = \phi_k P_k \phi_k^T - \phi_k P_k \phi_k^T M_{k+1}^T [M_{k+1} \phi_k P_k \phi_k^T M_{k+1}^T + R_{k+1}]^{-1} \cdot M_{k+1} \phi_k P_k \phi_k^T \tag{29}$$

Denote $\phi_k P_k \phi_k^T$ as the propagation of the error covariance P_k to time t_{k+1} before the measurement y_{k+1} is processed. Let the covariance at time t_k after the measurement y_k is processed be denoted as P_k^+ . Thus, letting $P_{k+1}^- = \phi_k P_k^+ \phi_k^T$ results in the following expression for the error covariance following the incorporation of the data at t_{k+1} .

$$P_{k+1}^+ = P_{k+1}^- - P_{k+1}^- M_{k+1}^T [M_{k+1} P_{k+1}^- M_{k+1}^T + R_{k+1}]^{-1} M_{k+1} P_{k+1}^- \quad (30)$$

Let the gain matrix be defined as follows:

$$K_{k+1} \triangleq P_{k+1}^+ M_{k+1}^T R_{k+1}^{-1} \quad (31)$$

For computational convenience the gain matrix can be expressed in terms of P_{k+1}^- by substitution of Equation (30) into Equation (31). The resulting recursive equations summarized below are the familiar Kalman optimal linear filter equations [1],

$$\begin{aligned} P_{k+1}^- &= \phi(t_{k+1}, t_k) P_k^+ \phi^T(t_{k+1}, t_k) \\ K_{k+1} &= P_{k+1}^- M_{k+1}^T [M_{k+1} P_{k+1}^- M_{k+1}^T + R_{k+1}]^{-1} \\ \hat{x}_{k+1} &= \phi(t_{k+1}, t_k) \hat{x}_k + K_{k+1} [y_{k+1} - M_{k+1} \phi(t_{k+1}, t_k) \hat{x}_k] \\ P_{k+1}^+ &= [I - K_{k+1} M_{k+1}^T] P_{k+1}^- \end{aligned} \quad (32)$$

V. A SECOND SPECIAL CASE - RECURSIVE LEAST SQUARES WITH CONSTRAINTS

Consider the problem now in which the unknown vector, x_k , is a constant and is constrained. That is, the constraints are given by

$$\underline{g}(x_k) \leq \underline{b}, \quad (33)$$

Let a sequence of measurements, which are corrupted by zero mean white noise, be related to the unknown parameter vector in a linear fashion as given in equation (5). Assume that the covariance of the measurement noise is unknown. Also assume that little or no information is known about the true value of the unknown vector. In this case, the problem reduces to simply choosing the parameter vector to best fit the data in a least squares sense subject to the constraint (33). From a computational standpoint, it is desirable to perform the estimation in a recursive fashion. Hence, this problem is a special case of the problem discussed in section III. The norm to be minimized is given by Equation (9) with the weighting matrix given in (10) reducing to the identity. Since the initial covariance of the states is unknown assume that the diagonal elements of P_0 approach infinity causing P_0^{-1} , denoted by D_0 , to approach the null matrix. Thus the results given by (23) can be specialized to this least squares data fitting problem. The two steps are:

- i) Update the cost functional $f(\hat{x}_{k+1})$
 1. $D_{k+1} = D_k + M_{k+1}^T M_{k+1}$ with $D_1 = M_1^T M_1$
 2. $c_{k+1} = c_k + 2M_{k+1}^T y_{k+1}$ with $c_1 = 2M_1^T y_1$
 3. $f(\hat{x}_{k+1}) = c_{k+1}^T \hat{x}_{k+1} - \hat{x}_{k+1}^T D_{k+1} \hat{x}_{k+1}$
- ii) Solve the nonlinear programming problem
maximize $f(\hat{x}_{k+1})$ subject to $\underline{g}(\hat{x}_{k+1}) \leq \underline{b}$

VI. STATISTICAL SIGNIFICANCE OF Q_K AND D_0

In the solution to the estimation problem in Section III the matrices

Q_k , D_0 and c_0 must be specified before the recursive algorithm can be implemented. As first noted by Ho [2] the deterministic and probabilistic approaches are closely related.

First assume that the measurement noise which causes y_k to be different than v_k has a joint gaussian density function.

$$p(v_k) = (2\pi)^{-\frac{m}{2}} |R_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} v_k^T R_k^{-1} v_k\right) \quad (35)$$

where v_k has zero mean and covariance R_k . Consider the measurements up to t_{k+1} and assume the v_i are statistically independent. Define,

$$R_{k+1}^{*-1} \triangleq \begin{bmatrix} R_1^{-1} & & 0 \\ & \ddots & \\ 0 & & R_{k+1}^{-1} \end{bmatrix} \quad \text{and note } |R_{k+1}^{*-1}| = \prod_{j=1}^{k+1} |R_j^{-1}|,$$

Letting v_{k+1}^* denote the augmented noise vector, the joint density becomes

$$p(v_{k+1}^*) = (2\pi)^{-\frac{(k+1)m}{2}} |R^*|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} J_{k+1}'\right) \quad (36)$$

where the functional J_{k+1}' is given by $\|v_{k+1}^*\|_{R_{k+1}^{*-1}}^2$.

Using the relation $v_{k+1}^* = \hat{y}_{k+1}^* - y_{k+1}^*$ and Equation (8) yields an expression for J_{k+1}' in terms of the estimate at t_{k+1} and the data.

$$J_{k+1}' = \left\| \begin{bmatrix} M_k^* & \phi_k^{-1} \\ & M_{k+1} \end{bmatrix} \hat{x}_{k+1} - \begin{bmatrix} y_k^* \\ v_{k+1} \end{bmatrix} \right\|_{R_{k+1}^{*-1}}^2 \quad (37)$$

The joint density function of (36) can be considered a likelihood function to be maximized. That is, it is desired to choose $\hat{x}_{k+1}$ subject to constraint (18) so as to minimize J_{k+1}' and thus maximize the density function. Comparison of equation (9) and (37) makes the following relation obvious. That is, $Q_k = R_k^{-1}$. The statistical interpretation of D_0 follows directly from the development of the Kalman equations in section IV. Since $\hat{x}_0$ was chosen as an interior point of the constraint set C, the system is unconstrained in a neighborhood of $\hat{x}_0$. Thus the development in section IV is valid in this neighborhood and hence D_0^{-1} is the estimation error covariance matrix at t_0 . The initial value of c_k can be determined from equation (24) as $c_0 = 2D_0 \hat{x}_0$.

VII. NUMERICAL EXAMPLE

A Monte Carlo analysis was performed on a simple example to illustrate the improvement possible when the constrained least squares (LS) algorithm is implemented. Assume the velocity of a missile immediately after being launched from an airplane can be modeled as

$$v(t) = a_0 t + v_0$$

where a_0 and v_0 are constants. Further, assume noisy velocity data is received every $\frac{1}{10}$ second by an interceptor and that an intercepting missile must be launched after four measurements are made. The value of the measurement noise variance is assumed to be unknown to the interceptor. The problem is to estimate a_0 and v_0 subject to the constraints that the maximum velocity of the launching airplane is 3000 ft/sec and the maximum acceleration of the missile is 676 ft/sec².

In the Monte Carlo simulation "actual" values of $a_0 = 620$ ft/sec² and $v_0 = 2700$ ft/sec were assumed. Measurements of missile velocity were made at 1, 2, 3 and 4 seconds. In generating the measurement noise in the simulation it was assumed that the opponent countermeasures could be modeled as zero mean white noise with a gaussian distribution. Runs were made for two sets of measurement noise variance. In the first the standard deviation was assumed to be 300 ft/sec or 7 1/2% of the 4000 ft/sec average velocity of the missile. The standard deviation of the noise on the second set of runs was 400 ft/sec. Seventy-five runs were made for each of three estimation methods and for each value of noise variance. The results are shown in Table 1. The standard LS method was used in order to get a comparison with analytical predictions which are shown in parentheses in Table 1.

In the second method the least squares estimate was modified such that if either a_0 or v_0 exceeded its maximum value that variable was set equal to that maximum. This is called limiting LS in Table 1. In the constrained LS algorithm Wolfe's quadratic programming method [3] was used. The improvement of constrained LS over limiting LS is dramatic when one variable exceeds the constraint by a significant margin. In one of the runs the standard LS gave estimates of 1062 ft/sec² and 1753 ft/sec for $\hat{a}_0$ and $\hat{v}_0$ respectively but the constrained LS method in limiting the estimate of $\hat{a}_0$ to 676 ft/sec² raised the estimate of $\hat{v}_0$ to 2717 ft/sec. It should be noted that the improvement in the rms error is dependent on the frequency with which the constraints are activated. This is, of course, a function of the signal noise ratio and the tightness of the constraints.

Since there was no need for intermediate estimates all four measurements were processed simultaneously. The determination of an estimate required only 1.6 seconds on the IBM 360/50 computer.

METHOD \ RMS MEAS NOISE	300 $\frac{\text{ft}}{\text{sec}}$		400 $\frac{\text{ft}}{\text{sec}}$	
	RMS(Δa_0) $\frac{\text{ft}}{\text{sec}^2}$	RMS(Δv_0) $\frac{\text{ft}}{\text{sec}}$	RMS(Δa_0) $\frac{\text{ft}}{\text{sec}^2}$	RMS(Δv_0) $\frac{\text{ft}}{\text{sec}}$
Standard Least Sq.	133 (134)	378 (366)	178 (178)	493 (488)
Limiting Least Sq.	106	314	115	422
Constrained Least Sq.	77	228	84	253
Percentage of times constraints activated	21%	31%	27%	35%

TABLE 1 RESULTS OF MONTE CARLO SIMULATION

VIII CONCLUSIONS

A computational algorithm for solving a special nonlinear filtering problem has been presented. In many problems, the solution yields optimal (minimum mean square error) estimates of the states of a linear system in which these states are constrained. The original problem was transcribed into a constrained minimum norm problem and recursive relations for the cost functional were derived. This rendered the problem tractable by permitting the use of efficient nonlinear programming methods. It is felt the algorithm has practical application in systems where the state variable constraints and the measurement noise are such that the constraints are activated frequently.

REFERENCES

1. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," ASME Journal of Basic Engineering, March, 1960.
2. Ho, Y. C., "The Method of Least Squares and Optimal Filtering Theory," Rand Corp. Memorandum RM-3329-PR.
3. Wolfe, P., "The Simplex Method for Quadratic Programming," *Econometrica*, 27, 1959, pp. 382-398.
4. Rosen, J., "The Gradient Projection Method for Nonlinear Programming, Part II, Nonlinear Constraints," *Journal of the Society for Industrial and Applied Mathematics*, 9, 1961, pp. 514-532.
5. Arrow, K. J., L. Hurwicz, and H. Uzawa, Studies in Linear and Non-linear Programming. Stanford, California: Stanford University Press, 1958.
6. Hadley, G., Nonlinear and Dynamic Programming. Reading, Mass: Addison-Wesley, 1964.

A STOCHASTIC MODEL-REFERENCE ADAPTIVE CONTROL SYSTEM*

S.H. WU and J.E. BRANDEBERRY
Department of Electrical Engineering
Marquette University
Milwaukee, Wisconsin

ABSTRACT

Stochastic Liapunov theory is employed in the design of a model reference adaptive control system. A controller is designed for step inputs, such that the plant converges in the mean norm to the model. Additional constraints are then imposed on the design of the controller to insure convergence with probability one.

SUMMARY

A number of physical systems may have time-varying parameters which cause the system response to differ from the desired response. Hiza and Li (1) investigated the design of an adaptive controller using deterministic Liapunov theory for a class of these systems, governed by linear time-varying differential equations with bounded coefficients. This paper extends their work to a class of systems governed by a stochastic differential equation with unbounded coefficients. For these systems deterministic Liapunov theory is inadequate; however controllers can be designed to provide convergence in the mean norm or with probability one using stochastic Liapunov theory.

The system to be controlled is a type-1 system governed by an n -th order differential equation with random time-varying coefficients and with unity feedback.

$$\text{System: } C^{(n)} + a_{n-1}(t) C^{(n-1)} + \dots + a_1(t) C^{(1)} = K(t) (e + e_a)$$

$$\text{where } e(t) = r(t) - C(t)$$

The coefficients of the system are assumed to be independent stationary random processes with independent increments and known first and second moments. The model is an $(n-1)$ -th order linear constant coefficient system with the following differential equation.

$$\text{Model: } C_d^{(n-1)} + b_{n-2} C_d^{(n-2)} + \dots + b_1 C_d^{(1)} + b_0 C_d = b_0 r(t)$$

The controller is then designed to provide convergence of the system to the model. The output of the controller $e_a(t)$ is permitted to be a function of the b_n 's, the states of the system and the model, the input $r(t)$, and the first and second moments of the coefficients of the system.

* This work was supported in part by the National Science Foundation under Grant No. GK-855.

Using the Liapunov theory developed by Bertram and Sarachik (2) a controller is designed for this system to provide convergence in the mean norm. The controller was found to be linear constant feedback of the states of the system. The feedback is a function of the coefficients of the model and the first moment of the system coefficients.

Then stochastic Liapunov theory developed by Kushner (3) is used to insure convergence with probability one. It is found that if $K(t)$ and $a_{n-1}(t)$ are constants and certain inequalities involving (P) and (Q) and the second moment of the $a_n(t)$'s can be satisfied, then the controller designed for convergence in the mean norm will also provide convergence with probability one. (P) and (Q) are the matrices related to the system and the model and the choice of a Liapunov function.

REFERENCES

- (1) Hiza, J.G. and C.C. Li, "On analytical Synthesis of a Class of Model-Reference Time-Varying Control Systems", IEEE Trans. on Applications and Industry Vol. 83, pp. 356-62, Nov. 1963.
- (2) Bertram, J.E. and P.E. Sarachik, "Stability of Circuits with Randomly Time-Varying Parameters", IRE Trans. on Circuit Theory CT-6, pp. 260-270, 1959.
- (3) Kushner, H.J., Stochastic Stability and Control, Academic Press, 1967.

AN APPROXIMATE METHOD OF OPTIMAL CONTROL
OF NONLINEAR DYNAMICAL SYSTEMS
WITH STATE-DEPENDENT NOISE

by

Yoshifumi Sunahara
Department of Mechanical Engineering
Faculty of Polytechnic Sciences
Kyoto Institute of Technology
Kyoto, Japan

1. Introduction

There is no need to say that dynamical systems to be controlled exhibit various kinds of nonlinear characteristics and may operate in a random environment whose stochastic characteristics undergo drastic changes. Thus, the general problem to be solved is to find the control of noisy nonlinear dynamical systems in some optimal fashion. Since the optimal control of nonlinear dynamical systems is almost impossible, in this paper, the author will introduce the reader to an approximate method which will be shown to play an important role in the realization of a broad class of stochastic optimal control.

2. Mathematical models and problem statement

Usually, the following stochastic integral equation of Ito-type is considered: [1][2]

$$(2.1) \quad \begin{aligned} x(t, \omega) = & x(\tau, \omega) + \int_{\tau}^t f[s, x(s, \omega)] ds \\ & + \int_{\tau}^t c(s) u(s) ds + \int_{\tau}^t g_D[s, x(s, \omega)] dw(s, \omega), \end{aligned}$$

where $x(t, \omega)$ is a Markov process, $w(t, \omega)$ is a Brownian motion process and ω is the generic point of the probability space Ω . We shall omit to write the symbol ω below because of no confusion, and use the same symbols for the true stochastic processes and for the quasi-linear stochastic processes which are the approximation to the true one by method described later. In Eq. (2.1), the function $f[t, x(t)]$ and $g_D[t, x(t)]$ are nonlinear functions, depending on both the state $x(t)$ and time t , and $u(t)$ is the control. Furthermore, we shall consider here only one-dimensional case for the convenience of discussions.

Instead of Eq. (2.1), however, we shall consider a new stochastic process determined by a stochastic integral equation with the diffusion term $D[x(t), t; \epsilon]$, where ϵ is a constant and this indicates the relative magnitude of the increment of the state-dependent Brownian motion process between t_0 and $T < \infty$. Thus, a concrete form of the diffusion term $D[x(t), t; \epsilon]$ is expressed by the state-independent portion $g_0(t)$ plus state-dependent portion $\epsilon g[x(t), t; \epsilon]$. From this point of view, the stochastic process with a state-dependent noise should be considered as $\{x(t), \epsilon, t \in [t_0, T]\}$ and this is given as a solution of the stochastic integral equation,

$$(2.2) \quad x(t) = x(\tau) + \int_{\tau}^t f[s, x(s)]ds + \int_{\tau}^t c(s)u(s)ds \\ + \int_{\tau}^t g_0(s)dw(s) + \varepsilon \int_{\tau}^t g[s, x(s)]dw(s).$$

The version of Eq. (2.2) into a stochastic differential equation is,

$$(2.3) \quad dx(t) = f[t, x(t)]dt + c(t)u(t)dt + g_0(t)dw(t) \\ + \varepsilon g[t, x(t)]dw(t).$$

In practical terms, the problem is to control $x(t)$ in such a way as to minimize a non-negative real valued functional,

$$(2.4) \quad J(u) = E\{\int_{t_0}^T L[t, x(t), u(t)]dt\} \quad [t_0 \leq t \leq T],$$

based on the a priori probability distribution of $x(t_0)$, provided that the process $x(s)$ for $t_0 \leq s \leq t$ is completely observable, where L is bounded, uniformly Hölder continuous in t and uniformly Lipschitz continuous in x . Let $\psi(t, \cdot)$ be a stochastic process, such that, for each $t \in [t_0, T]$,

$$(2.5) \quad \int_{t_0}^T E\{|\psi(t, \cdot)|^2\}dt < \infty.$$

Let Ψ denotes the class of $\psi(t)$ process. For some $\psi \in \Psi$, we call the control $u(t)$ admissible, if

$$(2.6) \quad u(t) = \psi(t, \cdot) \quad \text{for} \quad t \in [t_0, T].$$

For the security of mathematical development in the sequel, the following hypotheses are additionally made: [3]

H-1: The function $f[\cdot, \cdot]$ and $g[\cdot, \cdot]$ are Baire functions with respect to the pair (t, ξ) for $t_0 \leq t \leq T$ and $-\infty < \xi < \infty$, where $x(t) = \xi$.

H-2: The functions $f[\cdot, \cdot]$ and $g[\cdot, \cdot]$ satisfy uniform Lipschitz conditions in the variable ξ and are bounded by

$$(2.7) \quad |f[t, \xi]| \leq K_1(1 + \xi^2)$$

and

$$(2.8) \quad |g[t, \xi]| \leq K_2(1 + \xi^2),$$

where both K_1 and K_2 are real positive constants and are independent of both t and ξ respectively.

H-3: $x(t_0)$ is a random variable independent of the $w(t)$ -process.

With the property Eq. (2.6) and hypotheses H-1 to H-3, Eq. (2.3) has exactly a unique continuous solution $x(t)$. [4][5]

3. A quasi-linear stochastic differential equation

According to the hypotheses listed in section 2, the function $f[t, x(t)]$ may be expanded as [6]

$$(3.1) \quad f[t, x(t)] = a(t) + b(t)\{x(t) - \bar{x}(t|t)\} + e(t),$$

where $e(t)$ denotes the *error* terms. We shall determine $a(t)$ and $b(t)$ in such a way that the *mathematical* expectation of squared value of $e(t)$, $E\{e(t)^2\}$, becomes minimal with respect to $a(t)$ and $b(t)$. It is a simple exercise to show in the calculus of variation that the necessary and sufficient conditions for $\min E\{e(t)^2\}$ are given by

$$(3.2) \quad a(t) = E\{f[t, x(t)]\} \triangleq \bar{f}$$

and

$$(3.3) \quad b(t) = E\{[f[t, x(t)] - \bar{f}[t, x(t)]]\{x(t) - \bar{x}(t|t)\}\} p(t|t)^{-1}$$

where

$\bar{x}(t|t)$: expectation of $x(t)$

$p(t|t)$: covariance

i.e., $p(t|t) \triangleq \text{cov}[x(t)]$

We shall assume here that, for $t \in [t_0, T]$, the probability density function $p\{x(t)\}$, is Gaussian with the mean value $\bar{x}(t|t)$ and covariance $p(t|t)$, i.e.,

$$(3.4) \quad p\{x(t)\} = (2\pi)^{-\frac{1}{2}} p(t|t)^{-\frac{1}{2}} \exp[-\frac{1}{2}\{x(t) - \bar{x}(t|t)\}^2 p(t|t)^{-1}].$$

With the help of Eq. (3.4), both $a(t)$ and $b(t)$ can be obtained in the form $a(t) = a(t, \bar{x}(t|t), p(t|t))$ and $b(t) = b(t, \bar{x}(t|t), p(t|t))$. A striking fact is that the random variables $a(t)$ and $b(t)$ are not independent but depend mutually on the expectation value $\bar{x}(t|t)$ and covariance $p(t|t)$. From this point of view, in reality, more precise symbols, $a(t, \bar{x}(t|t), p(t|t))$ and $b(t, \bar{x}(t|t), p(t|t))$ should be introduced. However, for the economy of description, we merely denote these by $a(t)$ and $b(t)$ without indicating the dependence on both $\bar{x}(t|t)$ and $p(t|t)$.

We expand another nonlinear function $g[t, x(t)]$ in Eq. (2.3) into Taylor series. By neglecting the higher order terms more than $O(dt^{3/2})$, the differential $dg[t, x(t)]$ of the function $g[t, x(t)]$ becomes

$$(3.5) \quad \begin{aligned} dg[t, x(t)] &= \frac{\partial g[t, x(t)]}{\partial t} dt + f_q[t, x(t), \bar{x}(t|t)] \frac{\partial g[t, x(t)]}{\partial x(t)} dt \\ &\quad + \frac{1}{2} \frac{\partial^2 g[t, x(t)]}{\partial x(t)^2} \{g_0(t) + \varepsilon g[t, x(t)]\}^2 dt \\ &\quad + \frac{\partial g[t, x(t)]}{\partial x(t)} \{g_0(t) + \varepsilon g[t, x(t)]\} dw(t), \end{aligned}$$

where

$$(3.6) \quad f_q[t, x(t), \bar{x}(t|t)] \triangleq \bar{f}[t, x(t)] + b(t)\{x(t) - \bar{x}(t|t)\}.$$

Eq. (3.5) implies that $dg[t, x(t)]$ is a class of stochastic differentials and that, for $s \in [\tau, t]$, the stochastic integral equation of the following form is established:

$$(3.7) \quad g[t, x(t)] - g[\tau, x(\tau)]$$

$$= \int_{\tau}^t \left[\frac{\partial g[s, x(s)]}{\partial s} + f_q[s, x(s), \bar{x}(s|s)] \frac{\partial g[s, x(s)]}{\partial x(s)} + \frac{1}{2} \frac{\partial^2 g[s, x(s)]}{\partial x(s)^2} \right. \\ \left. \times \{g_0(t) + \varepsilon g[s, x(s)]\}^2 \right] ds + \int_{\tau}^t \frac{\partial g[s, x(s)]}{\partial x(s)} \{g_0(s) + \varepsilon g[s, x(s)]\} dw(s).$$

Specifically, we suppose that, for $\tau \leq s \leq t$, the interval $[\tau, t]$ is sufficiently small so as to make the integrands of the right side in Eq. (3.7) constants respectively. Hence it follows from Eq. (3.7) that

$$(3.8) \quad g[t, x(t)] - g[\tau, x(\tau)] \\ = \left[\frac{\partial g[\tau, x(\tau)]}{\partial \tau} + f_q[\tau, x(\tau), \bar{x}(\tau|\tau)] \frac{\partial g[\tau, x(\tau)]}{\partial x(\tau)} + \frac{1}{2} \frac{\partial^2 g[\tau, x(\tau)]}{\partial x(\tau)^2} \right. \\ \left. \times \{g_0(\tau) + \varepsilon g[\tau, x(\tau)]\}^2 \right] (t - \tau) + \frac{\partial g[\tau, x(\tau)]}{\partial x(\tau)} \{g_0(\tau) + \varepsilon g[\tau, x(\tau)]\} \\ \times \{w(t) - w(\tau)\}.$$

By applying Eqs. (3.1) and (3.8) to Eq. (2.3), we have

$$(3.9) \quad dx(t) = f_q[t, x(t), \bar{x}(t|t)]dt + c(t)u(t)dt + g_0(t)dw(t) \\ + \varepsilon [g[\tau, x(\tau)] + \gamma(\tau)(t - \tau) + \delta(\tau)\{w(t) - w(\tau)\}]dw(t),$$

where

$$(3.10) \quad \gamma(\tau) \triangleq \frac{\partial g[\tau, x(\tau)]}{\partial \tau} + f_q[\tau, x(\tau), \bar{x}(\tau|\tau)] \frac{\partial g[\tau, x(\tau)]}{\partial x(\tau)} \\ + \frac{1}{2} \frac{\partial^2 g[\tau, x(\tau)]}{\partial x(\tau)^2} \{g_0(\tau) + \varepsilon g[\tau, x(\tau)]\}^2$$

$$(3.11) \quad \delta(\tau) \triangleq \frac{\partial g[\tau, x(\tau)]}{\partial x(\tau)} \{g_0(\tau) + \varepsilon g[\tau, x(\tau)]\}.$$

Let $\phi(t, \tau)$ be the solution of the homogeneous differential equation $dx(t)/dt = b(t)x(t)$. The solution of Eq. (3.9) can formally be written as

$$(3.12) \quad x(t) = \phi(t, \tau)x(\tau) + \int_{\tau}^t \phi(t, s)[a(s) - b(s)\bar{x}(s|s)]ds \\ + \int_{\tau}^t \phi(t, s)c(s)u(s)ds + \int_{\tau}^t \phi(t, s)g_0(s)dw(s) \\ + \varepsilon \int_{\tau}^t \phi(t, s)\{g[\tau, x(\tau)] + \gamma(\tau)(s - \tau)\}dw(s) \\ + \varepsilon \int_{\tau}^t \phi(t, s)\delta(\tau)\{w(s) - w(\tau)\}dw(s).$$

Based on Eqs. (3.9) and (3.12), we have the followings after simple calculations,

$$(3.13) \quad d\bar{x}(t|t) = \bar{f}[t, x(t)]dt + c(t)\bar{u}(t)dt$$

and

$$(3.14) \quad \frac{dp(t|t)}{dt} = 2b(t)p(t|t) + 2c(t)k(t)p(t|t) + g_0(t)^2 + 2\varepsilon g_0(t)g[\tau, x(\tau)],$$

provided that the optimal control can be given in the form $u(t) = k(t)x(t) + r(t)$, and where $\bar{u}(t) = E[u(t)|x(t) = \xi]$.

4. Quasi-linear optimal control

Let the function L in Eq. (2.4) be

$$(4.1) \quad L[t, x, u] = m(t)x(t)^2 + n(t)u(t)^2,$$

where $m(t) > 0$ and $n(t) > 0$. In the case where the dynamical system is determined by linear stochastic differential, it has already been verified that the optimal control exists and is $u_0^*(t) = \psi_0^*[t, x(t)] = n(t)^{-1}c(t)q(t)x(t)$, where $q(t)$ is the unique solution of a certain Riccati type equation. [4][7] In the case of nonlinear regulator problem considered, the quasi-optimal control may be found out by an extensive use of the quasi-linearization technique developed in the previous section to the version of stochastic control. It is apparent that the $x(t)$ -process has the quasi-linear stochastic differential,

$$(4.2) \quad \begin{aligned} dx(t) = & b(t)x(t)dt + \{a(t) - b(t)\bar{x}(t|\tau)\}dt \\ & + c(t)\psi(t)dt + g_0(t)dw(t) + \varepsilon[g[\tau, x(\tau)] + \gamma(\tau)(t-\tau) \\ & + \delta(\tau)\{w(t) - w(\tau)\}]dw(t), \end{aligned}$$

where the definition of the admissible control given by Eq. (2.6) has been taken into account with the simplified notation $\psi(t)$.

In the present case, the cost rate function is given by Eq. (4.1) and performance index by Eq. (2.4).

Combining the stochastic linearization technique with the line of attack on the linear regulator problem, we shall suppose that $u(t) = \psi[t, x(t) = \xi]$. It has been proved by solving the following Bellman's equation [8] that the optimal control ψ_0^* and $V[t, \xi]$ exist.

$$(4.3) \quad \min_{\psi} \{ \bar{L}[t, \xi, \psi] + V_t[t, \xi] + \{f_q[t, \xi, \bar{\xi}] + c(t)\psi\}V_{\xi}[t, \xi] \\ + \frac{1}{2}\{g_0(t)^2 + 2\varepsilon g_0(t)g[\tau, x(\tau)]\}V_{\xi\xi}[t, \xi] \} = 0,$$

with terminal condition

$$(4.4) \quad V[T, \xi] = 0,$$

where

$$(4.5) \quad V[t, \xi] = E\left\{\int_t^T L[s, x(s), \psi^0[s, x(s)]]ds \mid x(t) = \xi\right\}$$

and

$$(4.6) \quad \bar{L}[t, \xi, \psi] = E\{L[t, \xi, \psi^0[t, \xi]|x(t) = \xi\}.$$

When the function L is given by Eq. (4.1), it follows from Eq. (4.6) that

$$(4.7) \quad \bar{L}[t, \xi, \psi(t, \xi)] = m(t)\xi^2 + n(t)\psi^2$$

We shall suppose that Bellman's equation Eq. (4.3) has a solution

$$(4.8) \quad V[t, \xi] = \pi(t)\xi^2 + 2\alpha(t)\xi + \beta(t),$$

where $\pi(t)$, $\alpha(t)$ and $\beta(t)$ will be determined as the solutions of differential equations which will be given later.

Applying Eqs. (4.7) and (4.8) to Eq. (4.3) and performing the minimization of Eq. (4.3), the optimal control is

$$(4.9) \quad \psi^0[t, \xi] = -n(t)^{-1}c(t)\pi(t)\xi - n(t)^{-1}c(t)\alpha(t),$$

and $\pi(t)$, $\alpha(t)$ satisfy

$$(4.10) \quad \frac{d\pi(t)}{dt} - n(t)^{-1}c(t)^2\pi(t)^2 + 2b(t)\pi(t) + m(t) = 0,$$

$$(4.11) \quad \frac{d\alpha(t)}{dt} + \{b(t) - n(t)^{-1}c(t)^2\pi(t)\}\alpha(t) + \{a(t) - b(t)\bar{x}(t|t)\}\pi(t) = 0,$$

for $t_0 \leq t \leq T$ with

$$(4.12) \quad \pi(T) = 0, \quad \alpha(T) = 0.$$

Furthermore, $\beta(t)$ in Eq. (4.8) satisfies

$$(4.13) \quad \frac{d\beta(t)}{dt} - n(t)^{-1}c(t)^2\alpha(t)^2 + 2\{a(t) - b(t)\bar{x}(t|t)\}\alpha(t) + \{g_0(t)^2 + 2\epsilon g_0(t)g[\tau, x(\tau)]\}\pi(t) = 0$$

for $t_0 \leq t \leq T$ with

$$(4.14) \quad \beta(T) = 0,$$

and this is necessary to compute Eq. (4.8), with $\pi(t)$ and $\alpha(t)$.

The optimal control depends on the cost rate function coefficient $m(t)$ and $n(t)$ and on the system dynamics $f[t, x(t)]$. However, a serious difficulty arises in the version of numerical computation on Eqs. (4.9), (4.10), (4.11), (4.12), (4.13) and (4.14). In fact the computation of Eq. (4.9) with Eqs. (4.10) and (4.11) has to start with the pre-assigned initial values of the state expectation $\bar{x}(t_0|t_0)$ and covariance $p(t_0|t_0)$ and, furthermore, with $\pi(t_0)$ and $\alpha(t_0)$ which are determined by the so-called trial and error method.

5. An illustrative example

For the purpose of exploring the quantitative aspects, we shall consider here the following. The dynamical system considered here is schematically shown by block diagram in Fig. 1. The stochastic differential equation of the dynamical system is given by

$$(5.1) \quad dx = f[-x]dt + udt + (c_1 + \epsilon x)dw$$

with

$$(5.2) \quad f[x] = \sin x,$$

where c_1 and ϵ are constants respectively. Application of Eq. (3.2) and (3.3) to the present case gives

$$(5.3) \quad a(t) = -\sin \bar{x} \exp(-0.5p)$$

and

$$(5.4) \quad b(t) = -\cos \bar{x} \exp(-0.5p).$$

From Eqs. (3.13) and (3.14), the approximated model and related covariance are determined by

$$(5.5) \quad d\bar{x} = -\sin \bar{x} \exp(-0.5p)dt + \bar{u}dt$$

and

$$(5.6) \quad \frac{dp}{dt} = -2p \cos \bar{x} \exp(-0.5p) - 2\pi(t)p + c_1^2 + 2\epsilon c_1 x(\tau).$$

Letting $m(t) = 1$ and $n(t) = 1$, in Eq. (4.1), we have

$$(5.7) \quad \psi^0[t, \xi] = -\pi(t)\xi - \alpha(t)$$

and

$$(5.8) \quad V[t, \xi] = \pi(t)\xi^2 + 2\alpha(t)\xi + \beta(t),$$

where

$$(5.9) \quad \frac{d\pi(t)}{dt} = \pi(t)^2 + 2\pi(t) \cos \bar{x} \exp(-0.5p) - 1$$

$$(5.10) \quad \begin{aligned} \frac{d\alpha(t)}{dt} &= (\sin \bar{x} - \bar{x} \cos \bar{x})\pi(t) \exp(-0.5p) \\ &+ \alpha(t)\{\pi(t) + \cos \bar{x} \exp(-0.5p)\} \end{aligned}$$

$$(5.11) \quad \begin{aligned} \frac{d\beta(t)}{dt} &= 2(\sin \bar{x} - \bar{x} \cos \bar{x})\alpha(t) \exp(-0.5p) + \alpha(t)^2 \\ &- (c_1^2 + 2\epsilon c_1 x(\tau))\pi(t). \end{aligned}$$

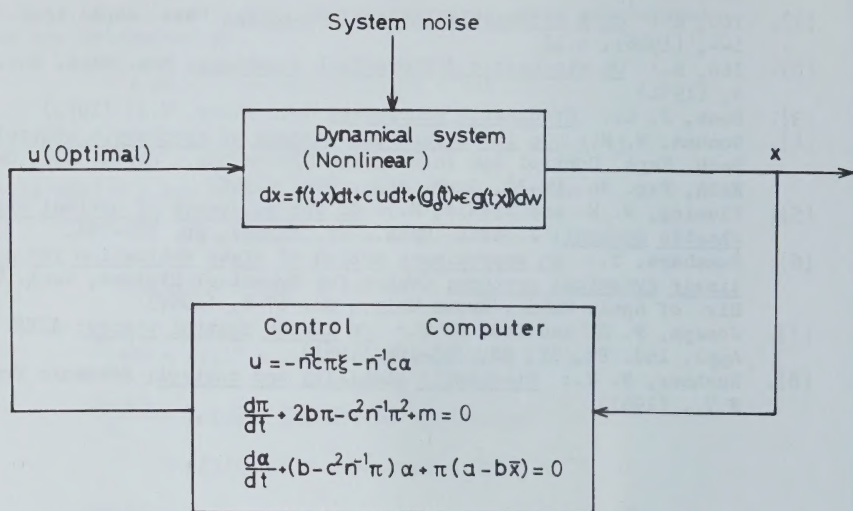
Equations (5.5) to (5.11) are simulated on a digital computer with the subroutine for the generation of random disturbance. Fig. 2(a) shows the sample path behavior of $x(t)$ and $u(t)$, for pre-assigned control interval $[0, 1.5](\text{sec})$. Fig. 2(b) shows the covariance $p(t|t)$, $\pi(t)$ and $\alpha(t)$ which may be adopted as a successful set of trial and error method.

6. Conclusion

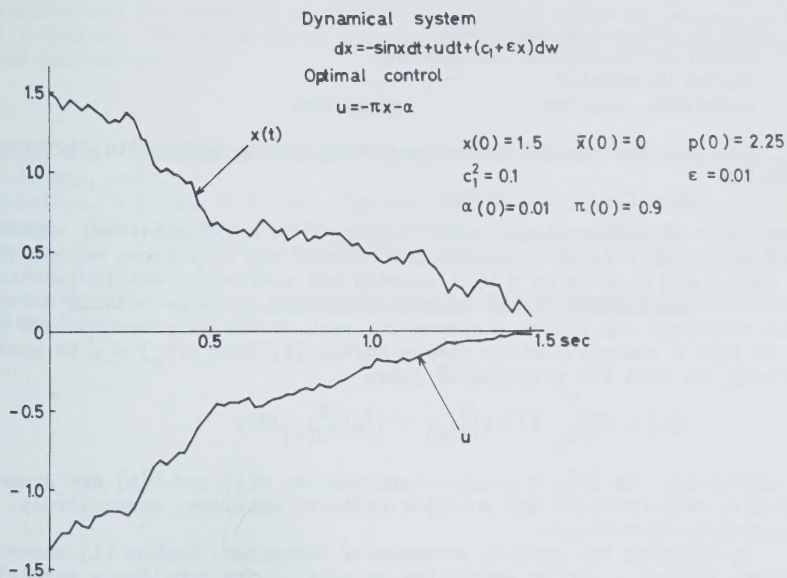
The technique started with the stochastic linearization of dynamical system. Then, the optimal control has been obtained for the linearized system by means of solving Bellman's equation. In general, the optimal control depends parametrically on both the conditioned averaged behavior $\bar{f}[t, x(t)] = a(t)$ of nonlinear action and the choice of performance index factors $m(t)$, $n(t)$. Through the analytical development and numerical results, we may conclude that the approximation procedure has desirable properties in realizing the feedback configuration of stochastic optimal control.

References

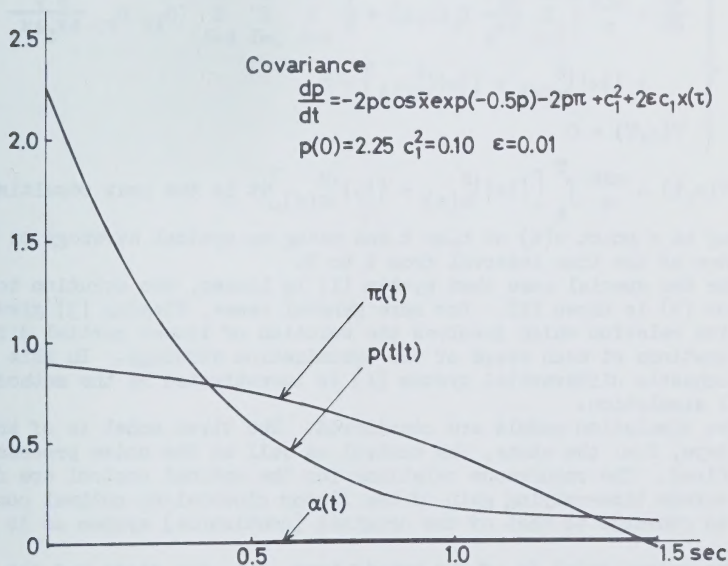
- [1]. Ito, K.: On a stochastic integral equation; Proc. Japan Acad. Nos. 1-4, (1946), p.32.
- [2]. Ito, K.: On stochastic differential equations; Mem. Math. Soc., 4, (1951)
- [3]. Doob, J. L.: Stochastic processes; John Wiley, N.Y. (1953)
- [4]. Wonham, W. M.: On the separation theorem of stochastic control; Tech. Reps. Control and Information System Lab., Elect. Res. Center, NASA, Rep. No. PM-38, Camb. Mass. Jan. (1968)
- [5]. Fleming, W. H. and Nishio, M.: On the existence of optimal stochastic control; J. Math. Mech., 15, (1966), pp. 774-794.
- [6]. Sunahara, Y.: An approximate method of state estimation for non-linear dynamical systems; Center for Dynamical Systems, Tech. Reps., Div. of Appl. Math., Brown Univ., No. 67-8, (1967)
- [7]. Joseph, P. D. and Tou, J. T.: On linear control theory; AIEE. Trans. Appl. Ind. Pt. II, 80, 193-196 (1961)
- [8]. Kushner, H. J.: Stochastic stability and control; Academic Press, N.Y., (1967)



[Fig. 1] Configuration of Optimal Control
For The Nonlinear Dynamical Systems
With State-dependent Noise



[Fig. 2(a)] Sample Path of Optimal Control



[Fig. 2(b)] Covariance and Feedback Gain

STOCHASTIC CONTROL SYSTEMS WITH STATE-DEPENDENT NOISE

J.Y.S. Luh and M. Maguiraga
School of Electrical Engineering
Purdue University
Lafayette, Indiana

Consider the control system described by the stochastic differential equation

$$dx = f(x,u)dt + G(x)dn \quad (1)$$

where x is an m -dimensional state vector; u is a k -dimensional control vector, $k \leq m$; f is an m -dimensional bounded and continuous vector function of both x and u ; G is an m by h bounded and continuous matrix function of x ; n is an h -dimensional Wiener process such that dn is gaussianly distributed with zero mean and finite variance for each of its components. The problem is to find a control u which steers system (1) from $x(t_0) = \bar{x}$ to some $x(T)$, T fixed, so that the performance index

$$J(u) = E \left\{ \int_{t_0}^T [\|x\|_{W(t)}^2 + \|u\|_{U(t)}^2] dt \right\} \quad (2)$$

is minimized. In (2), E denotes expectation, $W(t)$ and $U(t)$ are symmetric positive semi-definite and positive definite matrices, respectively, with appropriate dimensions.

By applying the dynamic programming technique, Wonham [1] shows that optimal control u can be determined by solving the non-linear partial differential equation

$$\begin{cases} \frac{\partial V}{\partial t} + \min_u \left[\sum_{i=1}^m \frac{\partial V}{\partial x_i} f_i(x,u) + \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^h G_{ik} G_{kj} \frac{\partial^2 V}{\partial x_i \partial x_j} \right. \\ \quad \left. + \|x\|_{W(t)}^2 + \|u\|_{U(t)}^2 \right] = 0 \\ V(x,T) = 0 \end{cases} \quad (3)$$

where $V(x,t) = \min_u \int_t^T [\|x\|_{W(t)}^2 + \|u\|_{U(t)}^2] dt$ is the cost resulting from starting at a point $x(t)$ at time t and using an optimal strategy on the remainder of the time interval from t to T .

For the special case that system (1) is linear, the solution to equation (3) is known [2]. For more general cases, Fleming [3] gives a recursive relation which involves the solution of linear partial differential equations at each stage of the optimization strategy. In this paper the stochastic differential system (1) is investigated by the method of digital simulation.

Two simulation models are considered. The first model is of the discrete type, i.e. the state, the control as well as the noise process are discretized. The recurrence relations for the optimal control are derived. The discrete time-varying gain of the linear closed-loop optimal control is shown to converge to that of the original (continuous) system as Δt approaches zero.

The second model is of the hybrid type, i.e. the state and the control remain to be continuous but the noise process is discretized. The linear closed-loop optimal control is determined. However, it does not converge to the optimal control of the original (continuous) system but to a limiting function. Thus some compensation terms are required in the hybrid model. These terms can be determined analytically.

The convergence proofs are based on Skorokhod's theorem [4]. The above conclusions are verified by numerical examples which are solved on a digital computer. The noise terms are simulated by using the pseudo-random generator subroutine.

REFERENCES

1. Wonham, W.M. "Stochastic Problems in Optimal Control," RIAS Tech. Report 63-14, May, 1963.
2. Ariaratnam, S.T. and N.K. Loh, "Optimal Control of Linear Stochastic Systems," International Journal of Control, Vol. 6, 1967, pp. 51-64.
3. Fleming, W.H. "Some Markovian Optimization Problems," J. Math. and Mech., Vol. 12, 1963, pp. 131-140.
4. Skorokhod, A.V. "Studies in the Theory of Random Processes", Addison-Wesley, 1965, pp. 157-161.

CONTINUOUS DELAY SYSTEMS UNDER DISCRETE CONTROL

SHELDON T. BECKER
Illinois Institute of Technology*
Chicago, Illinois

PHILIP V. LOPRESTI
Northwestern University
Evanston, Illinois

ABSTRACT

The necessary conditions for the optimal control of a general class of time lag systems, described by the differential-difference equation

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{x}(t-\theta), \underline{u}(t), t)$$

and under discrete control are established in the form of a local maximum principle.

INTRODUCTION

Piecewise constant control of continuous systems results when the computational solution of a control problem is obtained by use of a digital computer. The necessary conditions for optimality using this type of control were investigated for systems without time delays by Jordan and Polak¹. It is shown in this paper, that the results of Jordan and Polak can be extended to systems with time delays by use of the works of Kharatishvili². An important application of the results of this paper would be "on line" control of industrial processes.

PROBLEM DESCRIPTION

Find the necessary conditions for the minimization of the performance function $x_0(t_N)$ defined by the ordinary differential equation

$$\dot{x}_0 = g_0(x_1, \dots, x_n, u_1, \dots, u_m, t) \quad (1a)$$

subject to the constraints

$$\dot{x}_i = g_i(x_1, \dots, x_n, w_1, \dots, w_n, u_1, \dots, u_m, t) \quad (1b)$$

where the functions g_0 and g_i are differentiable in all arguments, and the variables, w , are defined as

$$w_0(t) \equiv 0 \quad (1c)$$

$$w_i(t) = x_i(t-\theta) \quad (1d)$$

$$\text{for } i = 1, \dots, n \quad (1e)$$

It is assumed that the time arguments of the variables $\underline{x}$, $\underline{w}$ and $\underline{u}$ are the same as the explicit time argument, t . The control is to be chosen from the set U :

$$u(t) \in U \quad (1f)$$

U is convex and the control u is piecewise constant:

$$u_j(t) = u_j(t_k) \quad j = 1, \dots, m \quad (1g)$$

$$t_k \leq t < t_{k+1} \quad k = 0, \dots, N-1 \quad (1h)$$

* Now with Bell Telephone Laboratories, Whippany, N. J.

The number of control intervals, N , and the sampling instants, t_k , are assumed fixed.

The following initial conditions are imposed

$$x_0(t_0) = 0 \quad (2a)$$

$$x_i(t_0) = x_{i0} \quad (2b)$$

and a continuous initial function on the components x_i ($i = 1, \dots, n$),

$$x_i(\tau) = \varphi_i(\tau) \quad (2c)$$

over the initial interval

$$t_0 - \theta \leq \tau < t_0 \quad (2d)$$

In order to be more concise, let

$$\underline{x} = (x_0, \dots, x_n) \quad (3a)$$

$$\underline{w} = (w_0, \dots, w_n) \quad (3b)$$

$$\underline{u} = (u_1, \dots, u_m) \quad (3c)$$

and

$$\underline{f}(\underline{x}, \underline{w}, \underline{u}, t) = [g_0(x_1, \dots, x_n, u_1, \dots, u_m, t), \dots, (g_n(x_1, \dots, x_n, w_1, \dots, w_n, u_1, \dots, u_m, t))]$$

Then

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{w}, \underline{u}, t) \quad (4)$$

represents a system F with initial conditions given by Eq. (2).

Adjoint System

Define an adjoint to F as:

$$\dot{\underline{p}} = - \frac{\partial \underline{f}'}{\partial \underline{x}}(\underline{x}, \underline{w}, \underline{u}, t) \underline{p} - \frac{\partial \underline{f}'}{\partial \underline{w}}(\underline{x}, \underline{w}, \underline{u}, t + \theta) \underline{v} \quad (5a)$$

$$\text{for } t_0 \leq t < t_N - \theta \quad (5b)$$

and

$$\dot{\underline{p}} = - \frac{\partial \underline{f}'}{\partial \underline{x}}(\underline{x}, \underline{w}, \underline{u}, t) \underline{p} \quad (5c)$$

$$\text{for } t_N - \theta \leq t \leq t_N \quad (5d)$$

The arguments of the variables $\underline{x}, \underline{p}$, etc., are the same as the explicit time argument of $\underline{f}$. The $n+1$ vectors $\underline{p}$ and $\underline{v}$ are defined in Eq. (6) as

$$\underline{p} = (p_0, \dots, p_n) \quad (6a)$$

$$\underline{v} = (v_0, \dots, v_n) \quad (6b)$$

and

$$v_i(t) = p_i(t + \theta) \quad (6c)$$

for

$$i = 0, \dots, n \quad (6d)$$

The prime indicates matrix transpose, and the $(n+1) \times (n+1)$ matrices of partials are defined as

$$\frac{\partial \underline{f}}{\partial \underline{x}} = \left[\frac{\partial f_i}{\partial x_j} \right] \quad (6e)$$

$$\frac{\partial \underline{f}}{\partial \underline{w}} = \left[\frac{\partial f_i}{\partial w_j} \right] \quad (6f)$$

The Hamiltonian

Define a Hamiltonian for the system as

$$H(\underline{x}, \underline{w}, \underline{p}, \underline{v}, \underline{u}, t_k + 1, t_k) = \langle \underline{p}(t_k + 1), \Delta \underline{x}(t_{k+1}, t_k) \rangle \quad (7)$$

where

$$\langle \underline{p}(t_k + 1), \Delta \underline{x}(t_k + 1, t_k) \rangle = \langle \underline{p}(t_k + 1), \Delta \underline{x}(t_k + 1, t_k) \rangle + \quad (8a)$$

$$\int_{t_k + 1 - \theta}^{t_k + 1} \langle \underline{p}(s + \theta), \underline{B}(s + \theta) \Delta \underline{x}(s, t_k) \rangle ds$$

where

$$\langle \underline{p}, \Delta \underline{x} \rangle = \underline{p}' \Delta \underline{x} \quad (8b)$$

and

$$\Delta \underline{x}(t, t_k) = \int_{t_k}^t \underline{f}(\underline{x}, \underline{w}, \underline{u}, s) ds \quad (8c)$$

The matrix $\underline{B}$ is defined as:

$$\underline{B} = \left[\frac{\partial f_i}{\partial w_j} \right] \quad (8d)$$

NECESSARY CONDITIONS FOR OPTIMALITY

Theorem

A necessary condition that $x_0(t_N)$ be a minimum is that
 (i) $H(\underline{x}^*, \underline{w}^*, \underline{p}^*, \underline{v}^*, \underline{u}^*, t_k + 1, t_k)$ is stationary or a relative maximum with respect to $\underline{u}$ along the optimal trajectories $\underline{x}^*, \underline{w}^*, \underline{p}^*, \underline{v}^*$.
 (ii) the optimal trajectories satisfy the sets of differential-difference equations (4) and (5) subject to the initial conditions of equation (2) and the terminal condition

$$\underline{p}^*(t_N) = (-1, 0, 0, \dots, 0) \quad (9)$$

Variational Equations

Introduce a perturbation in the optimal control

$$\underline{u}(t_k) = \underline{u}^*(t_k) + \epsilon \delta \underline{u}(t_k) \quad (10)$$

$$k = 0, \dots, N - 1 \quad (11)$$

where, ϵ and $\delta \underline{u}$ are chosen such that $\underline{u}$ is in U .

The resultant variation in $\underline{x}$ about the optimal trajectory, $\underline{x}^*$, depends continuously on t and ϵ for fixed $\delta \underline{u}$.

Note

$$\underline{x}^*(t) = \underline{x}(t, \epsilon) \Big|_{\epsilon = 0} \quad (12)$$

and

$$\underline{x}(t, \epsilon) = \underline{x}(t, 0) + \frac{\partial \underline{x}(t, 0)}{\partial \epsilon} \epsilon + O(\epsilon) \quad (13)$$

call

$$\frac{\partial \underline{x}(t, 0)}{\partial \epsilon} = \underline{y}(t) \quad (14)$$

It is straightforward to show that $\underline{y}(t)$ satisfies the linear system of differential-difference equations:

$$\dot{\underline{y}} = \frac{\partial \underline{f}}{\partial \underline{x}} \underline{y} + \frac{\partial \underline{f}}{\partial \underline{w}} \underline{z} + \frac{\partial \underline{f}}{\partial \underline{u}} \delta \underline{u} \quad (15)$$

Where the driving matrix is defined as

$$\frac{\partial \underline{f}}{\partial \underline{u}} = \left[\frac{\partial f_i}{\partial u_j} \right] \quad (16)$$

and

$$\delta \underline{u}(t) = \delta \underline{u}(t_k) \quad (17)$$

for

$$t_k \leq t < t_{k+1} \quad k = 0, 1, \dots, N-1 \quad (18)$$

The variable $\underline{z}$ is defined as the delayed value of $\underline{y}$, i.e.

$$\underline{z}(t) = \underline{y}(t - \theta) \quad (19)$$

This system (15) has a solution defined as follows

$$\begin{aligned} \underline{y}(t) = & \varphi(t, t_k) \underline{y}(t_k) + \int_{t_k}^t \varphi(t, s) \underline{B}(s) \underline{z}(s) ds + \\ & \int_{t_k}^t \varphi(t, s) \underline{C}(s) \delta \underline{u}(s) ds \end{aligned} \quad (20a)$$

Over the interval

$$t_k \leq t \leq t_k + \theta \quad (20b)$$

In succeeding intervals, $\underline{y}(t)$ is computed recursively according to the rule

$$\begin{aligned} \underline{y}(t) = & \varphi(t, t_k + m\theta) \underline{y}(t_k + m\theta) \\ & + \int_{t_k + m\theta}^t \varphi(t, s) \underline{B}(s) \underline{z}(s) ds \\ & + \int_{t_k + m\theta}^t \varphi(t, s) \underline{C}(s) \delta \underline{u}(s) ds \end{aligned} \quad (21a)$$

for successive integers $m = 1, 2, \dots$, and t lying in the interval

$$t \in [t_k + m\theta, t_k + (m+1)\theta] \quad (21b)$$

It follows immediately that $\underline{y}(t)$ is a linear functional of its initial function and of $\delta \underline{u}$ and a linear function of its initial state. Equivalently, $\underline{y}(t)$ may be considered a linear function of the $q-k+1$ vector variables $\delta \underline{u}(t_i)$ $i = k, \dots, q$, where q is the index for the sampling time, t_q , which satisfies the inequality

$$t_q \leq t < t_{q+1} \quad (22)$$

It also follows from the definition of $\underline{y}$, that if $\delta \underline{u}(t)$ is identically zero for $t < t_k$, then the initial conditions on $\underline{y}$ for a solution beginning at t_k are also identically zero. i.e.

$$\underline{y}(t_k) = 0 \quad (23a)$$

and
$$\underline{y}(t) \equiv 0 \quad (23b)$$

for
$$t_k - \theta \leq t < t_k. \quad (23c)$$

The solution for $\underline{y}$ is expressed in terms of

$\underline{\Phi}(t, t_k)$, the fundamental matrix solution of the differential equation

$$\dot{\underline{X}}(t) = \underline{A}(t) \underline{X}(t) \quad (24a)$$

with the initial condition

$$\underline{X}(t_k) = \underline{I} \quad (24b)$$

The matrices $\underline{A}$, $\underline{B}$ and $\underline{C}$, are the Jacobian matrices

$$\underline{A}(t) = \frac{\partial f(\underline{x}^*, \underline{u}^*, \underline{w}^*, t)}{\partial \underline{x}} \quad (25a)$$

$$\underline{B}(t) = \frac{\partial f(\underline{x}^*, \underline{u}^*, \underline{w}^*, t)}{\partial \underline{w}} \quad (25b)$$

$$\underline{C}(t) = \frac{\partial f(\underline{x}^*, \underline{u}^*, \underline{w}^*, t)}{\partial \underline{u}} \quad (25c)$$

Convexity of the variations

In accordance with the preceding discussion, the state variations are linear functions of $\delta \underline{u}(t), \dots, \delta \underline{u}(t_q)$, i.e. for the admissible perturbation $\epsilon \delta \underline{u}^1$, the variation $\epsilon \underline{y}$ at t is of the form

$$\epsilon \underline{y}^1(t) = \epsilon \sum_{i=k}^q \underline{h}_i(t, \delta \underline{u}^1(t_i)) \quad (26)$$

if $\epsilon \delta \underline{u}^1, \epsilon \delta \underline{u}^2$ and $\epsilon \delta \underline{u}^3$ are admissible where

$$\delta \underline{u}^3 = \delta \underline{u}^1 + \alpha \delta \underline{u}^2 \quad (27)$$

then it follows that

$$\underline{y}^3(t) = \sum_{i=k}^q \underline{h}_i(t, \delta \underline{u}^1(t_i) + \alpha \delta \underline{u}^2(t_i)) \quad (28a)$$

can be written as

$$\underline{y}^3(t) = \sum_{i=k}^q \underline{h}_i(t, \delta \underline{u}^1(t_i)) + \alpha \sum_{i=k}^q \underline{h}_i(t, \delta \underline{u}^2(t_i)) \quad (28b)$$

or

$$\underline{y}^3(t) = \underline{y}^1(t) + \alpha \underline{y}^2(t) \quad (28c)$$

Now, the particular perturbation $\epsilon \delta \underline{u}^3$ defined by the expression

$$\epsilon \delta \underline{u}^3 = \alpha \epsilon \delta \underline{u}^1 + (1 - \alpha) \epsilon \delta \underline{u}^2 \quad (29)$$

for

$$0 \leq \alpha \leq 1 \quad (30)$$

is easily shown to be admissible from the convexity of U , if $\epsilon \delta \underline{u}^1$ and $\epsilon \delta \underline{u}^2$ are admissible. Thus

$$\underline{y}^3(t) = \alpha \underline{y}^1(t) + (1 - \alpha) \underline{y}^2(t) \quad (31)$$

for any admissible $\epsilon \delta \underline{u}^1$ and $\epsilon \delta \underline{u}^2$ and for any t in the interval

$$t_k \leq t \leq t_N \quad (32)$$

Thus, the set of variations in $\underline{x}$ at t_N is convex. Furthermore, since

$$\delta \underline{u} \equiv 0 \quad (33)$$

is obviously admissible, this terminal set is a convex cone.

Relation between variation and adjoint system states

Consider a perturbation $\epsilon \delta \underline{u}^k$ in $\underline{u}$ for just one interval of time, i.e.

$$\delta \underline{u}^k(t) \neq 0 \quad (34)$$

only if

$$t \in [t_k, t_{k+1}]$$

then for t greater than t_{k+1} ,

$$\dot{\underline{y}}(t) = \underline{A}(t) \underline{y}(t) + \underline{B}(t) \underline{y}(t-\theta) \quad (35)$$

Now for $k = 0, \dots, N-1$, the inner product of $\underline{p}$ and $\underline{y}$ at t_N and t_{k+1} are equal as is shown by the following derivation.

$$\begin{aligned} & (\underline{p}(t_N), \underline{y}(t_N)) - (\underline{p}(t_{k+1}), \underline{y}(t_{k+1})) \\ &= \int_{t_{k+1}}^{t_N} \frac{d}{ds} (\underline{p}, \underline{y}) ds \end{aligned} \quad (36)$$

$$\begin{aligned} &= \int_{t_{k+1}}^{t_N} \frac{d}{ds} [\langle \underline{p}(s), \underline{y}(s) \rangle + \\ & \int_{s-\theta}^s \langle \underline{p}(\alpha + \theta), \underline{B}(\alpha + \theta) \underline{y}(\alpha) \rangle d\alpha] ds \end{aligned} \quad (37)$$

$$\begin{aligned} &= \int_{t_{k+1}}^{t_N} \frac{d}{ds} \langle \underline{p}(s), \underline{y}(s) \rangle ds + \\ & \int_{t_{k+1}}^{t_N} \frac{d}{ds} \int_{s-\theta}^s \langle \underline{p}(\alpha + \theta), \underline{B}(\alpha + \theta) \underline{y}(\alpha) \rangle d\alpha ds \end{aligned} \quad (38)$$

$$\begin{aligned}
&= \int_{t_{k+1}}^{t_N} \left\langle \frac{dp}{ds}, \underline{y} \right\rangle ds + \int_{t_{k+1}}^{t_N} \left\langle \underline{p}, \frac{dy}{ds} \right\rangle ds \\
&+ \int_{t_{k+1}}^{t_N} \langle \underline{p}(s + \theta), \underline{B}(s + \theta) \underline{y}(s) \rangle ds - \\
&\int_{t_{k+1}}^{t_N} \langle \underline{p}(s), \underline{B}(s) \underline{y}(s - \theta) \rangle ds \quad (39)
\end{aligned}$$

By substitution, it follows that

$$\begin{aligned}
&(\underline{p}(t_N), \underline{y}(t_N)) - (\underline{p}(t_k + 1), \underline{y}(t_k + 1)) = \\
&- \int_{t_{N-\theta}}^{t_N} \langle \underline{A}' \underline{p}, \underline{y} \rangle ds - \int_{t_{k+1}}^{t_{N-\theta}} \langle \underline{A}' \underline{p}, \underline{y} \rangle + \langle \underline{B}'(s + \theta) \underline{y}, \underline{y} \rangle ds \\
&+ \int_{t_{k+1}}^{t_N} \langle \underline{p}, \underline{A} \underline{y} \rangle + \langle \underline{p}, \underline{B}(s) \underline{z} \rangle ds \\
&+ \int_{t_{k+1}}^{t_N} \langle \underline{p}(s + \theta), \underline{B}(s + \theta) \underline{y} \rangle ds - \int_{t_{k+1}}^{t_N} \langle \underline{p}, \underline{B}(s) \underline{y}(s - \theta) \rangle ds \quad (40)
\end{aligned}$$

Where, if the argument of a term is omitted, it is understood to be s .
Cancelling common terms gives

$$\begin{aligned}
&(\underline{p}(t_N), \underline{y}(t_N)) - (\underline{p}(t_k + 1), \underline{y}(t_k + 1)) \\
&= \int_{t_{N-\theta}}^{t_N} \langle \underline{p}(s + \theta), \underline{B}(s + \theta) \underline{y}(s) \rangle ds \quad (41)
\end{aligned}$$

Now define

$$\underline{B}(s + \theta) \equiv 0 \quad (42)$$

$$\text{for } s > t_N - \theta \quad (43)$$

Hence

$$(\underline{p}(t_N), \underline{y}(t_N)) - (\underline{p}(t_k + 1), \underline{y}(t_k + 1)) = 0 \quad (44)$$

Proof of Theorem

Lemma 10 by Pontryagin, et al,³ applies directly to the proof here, and shows that the convex cone formed by the variations in $\underline{x}(t_N)$ lies entirely above the plane in $n + 1$ space defined by the equation

$$x_o = x_o^* (t_N) \quad (45)$$

The unit normal to this plane is

$$\underline{a} = (-1, 0, \dots, 0) \quad (46)$$

and

$$(\underline{a}, \epsilon \underline{y}(t_N)) \leq 0 \quad (47)$$

from the geometry of the situation.

Set

$$p^*(t_N) = \underline{a} \quad (48)$$

then, substituting into Eq. (47) gives

$$(p^*(t_N), \epsilon \underline{y}(t_N)) \leq 0 \quad (49)$$

but from Eq. (44), this implies

$$(p^*(t_k + 1), \epsilon \underline{y}(t_k + 1)) \leq 0 \quad (50)$$

Now

$$\Delta \underline{x}(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \underline{f}(\underline{x}(u^*(t_k) + \epsilon \delta \underline{u}^k, s) \underline{w}(u^*(t_k) + \epsilon \delta \underline{u}^k, s), \underline{u}^*(t_k) + \epsilon \delta \underline{u}^k, s) ds \quad (51)$$

The dependence of $\Delta \underline{x}$ on the initial conditions for $\underline{x}$ is also implicit in the above.

Now, under the perturbation chosen, $\Delta \underline{x}$ can also be expressed as:

$$\Delta \underline{x}(t_k + 1, t_k) = \underline{x}(t_k + 1, \epsilon) - \underline{x}^*(t_k) \quad (52)$$

Thus, dropping the arguments in the integrand; and substituting Eq. (52) in Eq. (51)

$$\underline{x}(t_{k+1}, \epsilon) - \underline{x}^*(t_k) = \int_{t_k}^{t_{k+1}} \underline{f}(\underline{x}, \underline{w}, \underline{u}, s) ds \quad (53)$$

Differentiating Eq. (53) and setting $\epsilon = 0$ gives

$$\left. \frac{\partial \underline{x}}{\partial \epsilon}(t_{k+1}, \epsilon) \right|_{\epsilon=0} = \frac{d}{d\underline{u}(t_k)} \int_{t_k}^{t_{k+1}} \underline{f}(\underline{x}, \underline{w}, \underline{u}, s) ds \left. \frac{d\underline{u}}{d\epsilon} \right|_{\epsilon=0} \quad (54)$$

The total derivative operator is used to emphasize that the initial conditions on $\underline{x}$ and the sampling instants t_k and t_{k+1} are fixed.

Thus

$$\underline{y}(t_k + 1) = \frac{d}{d\underline{u}(t_k)} \Delta \underline{x}^*(t_k + 1, t_k) \delta \underline{u}^k \quad (55)$$

Substituting into Eq. (49) gives

$$(p^*(t_k + 1), \frac{d}{d\underline{u}(t_k)} \Delta \underline{x}^*(t_k + 1, t_k) \epsilon \delta \underline{u}^k) \leq 0 \quad (56)$$

and recognizing that p^* is independent of $\underline{u}$ the desired result is obtained:

$$\left\langle \frac{\partial H^*}{\partial u}, \epsilon \delta u^k \right\rangle \leq 0 \quad (57)$$

Thus, part (i) of the theorem is proved. The proof of part (ii) of the theorem follows from the constructive nature of the procedure.

CONCLUSION

The necessary conditions have been derived for the optimal piecewise constant control of a continuous system with time delays and a free terminal condition. The extension to cases where the terminal constraints are a manifold or a point is relatively simple. In this paper, the control intervals were held constant; a further paper will consider the effect of allowing the time intervals to vary.

REFERENCES

- (1) B. Jordan and E. Polak, "Optimal Control of Aperiodic Discrete Time Systems", SIAM Journal on Control, Vol. 2, Jan. 1965, pp. 332-46.
- (2) G. Kharatishvili, "The Maximum Principle in the Theory of Optimal Processes With Delay", Doklady Akad. Nauk S.S.S.R., Vol. 136, 1961, pp. 39-42.
- (3) L. S. Pontryagin et al, The Mathematical Theory of Optimal Processes, New York: John Wiley and Sons, 1962.

DONG HAK CHYUNG
Department of Electrical Engineering
University of Iowa
Iowa City, Iowa 52240

ABSTRACT

The problem of finding an optimal control in linear discrete systems with time delays in both the state variables and control variables is studied.

The problem of finding an optimal control in linear discrete control systems with time delays in the state variables has been studied in [1]. In some cases the delays occur not only in the state variables but also in the control variables. In this paper the results in [1] are modified so as to include the case when the control variables also contain delays.

Consider the system

$$x(k+1) = A_0(k)x(k) + A_1(k)x(k-h_1) + B_0(k)u(k) + B_1(k)u(k-h_2) \quad (1)$$

with an initial condition $x(k) = \phi(k)$, $k = -h_1, -h_1+1, \dots, 0$. Here x and u are n and m dimensional vectors respectively, $A_0(k)$, $A_1(k)$, $B_0(k)$ and $B_1(k)$ are matrices with compatible dimensions. We assume that $0 < h_2 \leq h_1$. However this is purely arbitrary, and the case when $h_1 \leq h_2$ can be handled by the same manner. The case when there are multiple delays can also be handled by the same manner.

The usual optimal control problem is to find a control sequence $u = \{u(-h_2), u(-h_2+1), \dots, u(N-1)\}$ from a given control restraint set Ω which steers the corresponding response $x(k)$ of the system to a given target set G at $k=N$ and minimizes a given cost functional $C(u)$. Here N is a given fixed positive integer.

In this paper we assume that Ω is compact convex set in the m -dimensional real space R^m and the target G is the whole n -dimensional real space R^n (i.e., free endpoint problem). Other cases such as $\Omega = R^m$ or $G \subset R^n$ can be handled by a similar manner as in [2].

Suppose the cost functional $C(u)$ is given by

$$C(u) = g(x(N)) + \sum_{k=0}^{N-1} \{f(x(k), k) + h(u(k), k)\} \quad (2)$$

where $g(x)$ is a scalar C^1 convex function in x , $f(x, k)$ is a scalar non-negative C^1 convex function in x for each k , and $h(u, k)$ is a scalar non-negative continuous convex function in u for each k . Then the problem is now to find a control $u^* = \{u^*(-h_2), u^*(-h_2+1), \dots, u^*(N-1)\} \subset \Omega$ with corresponding response $x^*(k)$ which minimizes the cost functional $C(u)$.

The response of the system given by Eq (1) can be written as

$$x(k) = x(k, \phi) + \sum_{j=0}^{k-1} W(k, j+1) \{B_0(j)u(j) + B_1(j)u(j-h_2)\} \quad (3)$$

where $x(k, \phi)$ is the solution of

$$x(k+1) = A_0(k)x(k) + A_1(k)x(k-h_1), \quad k \geq 0$$

$$x(k) = \phi(k), \quad k = -h_1, -h_1+1, \dots, 0.$$

and $W(k, j)$ is an $n \times n$ matrix solution of

$$W(k, j) = \begin{cases} I, \text{ the } n \times n \text{ identity matrix} & , \text{ if } j = k \\ W(k, j+1)A_0(j) & , \text{ if } k-h_1 \leq j < k \\ W(k, j+1)A_0(j) + W(k, j+h_1+1)A_1(j+h_1), & \text{ if } 0 \leq j < k-h_1 \end{cases}$$

$W(k, j)$ is solved backwards starting from $j = k$.

Let $x^0(N) = \sum_{k=0}^{N-1} \{f(x(k), k) + h(u(k), k)\}$ and define an $(n+1)$ dimensional

vector $\hat{x}(N) = (x^0(N), x(N))$, where $x(N)$ is given by Eq (3). Then each control u determines a unique $\hat{x}(N)$ and the corresponding cost functional is $C(u) = g(x(N)) + x^0(N)$. Let $\hat{K}$ be the set of $\hat{x}(N)$ for all controls u in Ω . Then, since $\hat{x}(N)$ can be regarded as a continuous function of $u(-h_2)$, $u(-h_2+1), \dots, u(N-1)$, $\hat{K}$ is a continuous mapping of the compact set $\Omega \times \Omega \times \dots \times \Omega \subset \mathbb{R}^{(N+h_2)m}$. Therefore $\hat{K}$ is compact in $\mathbb{R}^{n+1}$.

The problem is now equivalent to finding a point in $\hat{K}$ where $C(u) = g(x) + x^0$ is minimum for all $\hat{x} = (x^0, x)$ in $\hat{K}$. The corresponding control is then an optimal control. Since $\hat{K}$ is compact and $C(u)$ is continuous in $\hat{x}$, there exists a point $\hat{x}^* = (x^{0*}, x^*)$ in $\hat{K}$ where $C(u)$ is minimum over $\hat{K}$. Therefore there always exists an optimal control. Furthermore, since $C(u)$ is a strictly increasing function in x^0 , $\hat{x}^*$ is a lowest boundary point of $\hat{K}$ in the x^0 -axis direction.

By a similar method as in [1] we can show that a control $u^* = \{u^*(-h_2), u^*(-h_2+1), \dots, u^*(N-1)\} \subset \Omega$ steers the corresponding response $\hat{x}^*(k)$ to the lowest boundary point of $\hat{K}$, i.e., u^* is an optimal control, if

and only if the control u^* satisfies the following condition.

$$q(k+h_2+1)B_1(k+h_2)u^*(k) = \max_{u \in \Omega} q(k+h_2+1)B_1(k+h_2)u, \quad -h_2 \leq k < 0$$

$$q_0 h(u^*(k), k) + q(k+1)B_0(k)u^*(k) + q(k+h_2+1)B_1(k+h_2)u^*(k)$$

$$= \max_{u \in \Omega} q_0 h(u, k) + q(k+1)B_0(k)u + q(k+h_2+1)B_1(k+h_2)u, \quad 0 \leq k < N-h_2$$

$$q_0 h(u^*(k), k) + q(k+1)B_0(k)u^*(k) = \max_{u \in \Omega} q_0 h(u, k) + q(k+1)B_0(k)u, \quad N-h_2 \leq k < N$$

where q_0 is a negative constant and $q(k)$ is a row vector solution of

$$q(k) = \begin{cases} q(k+1)A_0(k) + q(k+h_1+1)A_1(k+h_1) + q_0 \frac{\partial f}{\partial x}(x^*(k), k), & 0 \leq k < N-h_1 \\ q(k+1)A_0(k) + q_0 \frac{\partial f}{\partial x}(x^*(k), k), & N-h_1 \leq k < N \end{cases}$$

with endpoint condition $q(N) = -\text{grad } g(x^*(N))$.

In certain cases $u(k)$ for $k = -h_2, -h_2+1, \dots, -1$ is given by the system and hence can not be chosen to satisfy the above maximum condition for $-h_2 \leq k < 0$. In this case the maximum condition should be ignored for $k = -h_2, -h_2+1, \dots, -1$.

REFERENCES

1. D. H. Chyung, "Discrete Systems with Time Delay", The 5th Allerton Conference on Circuit and System Theory, University of Illinois, October 1967.
2. D. H. Chyung, "Discrete Linear System with Convex Cost Functional", The 4th Allerton Conference on Circuit and System Theory, University of Illinois, October 1966.

OPTIMIZATION OF DISCRETE-TIME SYSTEMS BY SUCCESSIVE APPROXIMATION

M. H. RICHARDSON AND R. J. LEAKE

College of Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

As an alternative to a direct solution of Bellman's recursive equation results are presented for the determination and improvement of sub-optimal controls, and for obtaining bounds on the optimum value of the performance index to provide a gauge for numerical experimentation. Also, methods for constructing both monotone increasing and decreasing sequences converging to the optimal value function are given.

INTRODUCTION

It is well known that direct attempts to solve Bellman's recursive equation in order to obtain optimal feedback control laws lead to computational difficulties. In order to limit the range of search and storage we propose methods for obtaining bounds on the value function. The approach taken is to a large extent, the discrete counterpart of that used in [1]. Rather than use the principle of optimality explicitly we employ a Kalman-Caratheodory [2] approach to establish our starting framework. It is worth noting that in this development the system state is not restricted to any particular kind of set; nor are the control inputs.

Consider a dynamical system represented by

$$x(k+1) = f(x(k), v(x(k), k), k) \quad x(k_0) = x_0 \quad (1)$$

where x belongs to an arbitrary state set X and $v: X \times K \rightarrow U$ is a feedback control function and K denotes the integers. The solution of (1) is denoted as $\varphi_v(k) = \varphi_v(k; x_0, k_0)$. Let the motion space G be a subset of $X \times K$ to which all motion $(\varphi_v(k), k)$ is restricted and let the target set S be a subset of G . The class $\mathcal{K}^0$ of admissible feedback control laws is defined such that $v \in \mathcal{K}^0$ IFF

- the values $v(x, k)$ of v belong to a preassigned set U for all (x, k)
- when v is substituted into (1), any motion beginning in G - S reaches S or approaches S in a uniform asymptotic manner without leaving G .

The terminal time $k_1(x_0, k_0; v)$ is the least integer $k_1 \geq k_0$ such that $(\varphi_v(k_1), k_1) \in S$; or, in the asymptotic case, $k_1 = \infty$.

The system performance is evaluated by

$$J(x_0, k_0; v) = \lambda(\varphi_v(k_1), k_1) + \sum_{\alpha=k_0}^{k_1-1} L(\varphi_v(\alpha), v(\varphi_v(\alpha), \alpha), \alpha) \quad (2)$$

where λ and L are real valued functions. An optimal control law is a function from the class $\mathcal{K}^0$ which minimizes J . Assuming the existence of such a function, we define

$$V^0(x, k) = \min_{v \in \mathcal{K}^0} J(x_0, k_0; v) \quad (3)$$

For any function $V: X \times K \rightarrow R$ (real numbers) we define H as

$$H(x, V, k, u) = V(f(x, u, k), k+1) - V(x, k) + L(x, u, k) \quad (4)$$

where $u \in U$. Let $\mathcal{J}(x, k) = \{u \in U | (f(x, u, k), k+1) \in G\}$. Assume that H has an absolute minimum with respect to all $u \in \mathcal{J}(x, k)$ and also that $u^0 = C(x, V, k)$ uniquely specifies the minimum. (If the minimum occurs at more than one point of $\mathcal{J}$, C will single out just one of the minima.)

We then define H^0 as

$$H^0(x, V, k) = \min_{u \in \Omega(x, k)} H(x, V, k, u) = H(x, V, k, C(x, V, k)) \quad (5)$$

Note that Bellman's equation is

$$H^0(x, V, k) = 0 \quad \text{or} \quad V(x, k) = \min_{u \in \Omega(x, k)} [L(x, u, k) + V(f(x, u, k), k+1)] \quad (6)$$

with boundary condition $V(x, k_1) = \lambda(x, k_1)$.

The solution of (1) is easily seen to have the property that for $\alpha \geq k \geq k_0$

$$\varphi_V(\alpha; \varphi_V(k; x_0, k_0), k) = \varphi_V(\alpha; x_0, k_0) = \varphi_V(\alpha) \quad (7)$$

and furthermore, if $(x_0, k_0) \in G-S$ we have for the terminal time

$$k_1(\varphi_V(k), k; v) = k_1(x_0, k_0; v) = k_1(v) \quad (8)$$

It follows from (2), (7), (8) that for $k_1 - k_0 \leq k \leq k_1$

$$J(\varphi_V(k), k; v) = \lambda(\varphi_V(k_1), k_1) + \sum_{\alpha=k}^{k_1} L(\varphi_V(\alpha), v(\varphi_V(\alpha), v(\varphi_V(\alpha), \alpha), \alpha) \quad (9)$$

Consequently, the "Eulerian" forward difference of J along motions of (1) is given by

$$\Delta J(\varphi_V(k), k; v) = -L(\varphi_V(k), v(\varphi_V(k), k), k) \quad (10)$$

or in a more casual notation

$$\Delta J(x, k; v) = -L(x, v, k) \quad (11)$$

Similarly, we may write the forward difference of a function $V: X \times K \rightarrow R$ as

$$\Delta V(\varphi_V(k), k) = V(\varphi_V(k+1), k+1) - V(\varphi_V(k), k) \quad (12)$$

or simply as

$$\Delta V(x, k) = V(f(x, v, k), k+1) - V(x, k) \quad (13)$$

In the interest of simpler notation, expressions (11) and (13) will be used instead of (10) and (12).

BASIC TRANSFORMATIONS AND THEORETICAL PRELIMINARIES

We define $\mathcal{V}$ as the set of all functions $V: X \times K \rightarrow R$ such that $V(x, k) = \lambda(x, k)$ on S , and define $\mathcal{V}^0$ as the subset of $\mathcal{V}$ for which $V \in \mathcal{V}^0$ implies $C(x, V, k)$ exists and is admissible.

We define the following transformations

- $T_1: \mathcal{V}^0 \rightarrow \mathcal{K}^0$ is defined for any $V \in \mathcal{V}^0$ by $T_1(V) = v$ where $v(x, k) = C(x, V, k)$
- $T_2: \mathcal{K}^0 \rightarrow \mathcal{V}$ is defined by $T_2(v) = V$ where $V(x, k) = J(x, k; v)$
- $T: \mathcal{V}^0 \rightarrow \mathcal{V}$ is the composite mapping defined for $V \in \mathcal{V}^0$ by $T(V) = T_2(T_1(V)) = J(x, k; v)$ with $v(x, k) = C(x, V, k)$.

Lemma 1. Let $W \in \mathcal{V}$ and $v \in \mathcal{K}^0$, $V \in \mathcal{V}^0$. Then

- $W = T_2(v)$ IFF $H(x, W, k, v) = 0$ in $G-S$
- $W = T(V)$ IFF $H(x, W, k, C(x, V, k)) = 0$ in $G-S$

proof: a) Assume $H(x, W, k, v) = 0$. Taking the forward difference of W along motions of (1) with control v we have from (4), (13)

$$\Delta W = H(x, W, k, v) - L(x, v, k) = -L(x, v, k)$$

Thus

$$J(x, k, v) = W(x_1, k_1) + \sum_{\alpha=k}^{k_1-1} \Delta W = W(x, k)$$

Conversely, from (11) $W(x, k) = J(x, k; v)$ implies $\Delta W = -L(x, v, k)$ which leads to $H(x, W, k, v) = 0$. b) follows from a) by setting $v = C$.

This result gives an alternate method of evaluating the performance index for a given admissible control law.

At this point we note that it is not at all obvious (contrary to the blithe assertions of [1]) that $V^0 \in \mathcal{V}^0$. Actually the next result says even more.

Lemma 2. a) $V^0 \in \mathcal{V}^0$, b) $C(x, V^0, k)$ exists and is an optimal feedback control law, c) $V^0 = T(V^0)$.

proof: Firstly, it is clear that $V^0(x, k) = \lambda(x, k)$ on S . We know V^0 exists. Therefore, pick $v^0(x, k) \in \mathcal{K}^0$ such that

$$V^0(x, k) = J(x, k; v^0) \quad (15)$$

Obviously v^0 is admissible. Let us now show that we may choose $C(x, V^0, k) = v^0(x, k)$. To do this we must show that for each $(x, k) \in G-S$ and $u \in \Omega(x, k)$

$$H(x, V^0(x, k), k, v^0(x, k)) \leq H(x, V^0(x, k), k, u)$$

By (15) and Lemma 1(a) we have

$$H(x, V^0(x, k), k, v^0(x, k)) = 0$$

Let us assume there exists $(x^*, k^*) \in G$ and $u^* \in \Omega(x^*, k^*)$ such that

$$H(x^*, V^0(x^*, k^*), k^*, u^*) < 0$$

that is $V^0(f(x^*, u^*, k^*), k^* + 1) + L(x^*, u^*, k^*) < V^0(x^*, k^*)$. This leads to an immediate contradiction, however, since the latter expression implies that the control law

$$v^*(x, k) = \begin{cases} v^0(x, k) & , (x, k) \neq (x^*, k^*) \\ u^* & , (x, k) = (x^*, k^*) \end{cases}$$

satisfies $J(x^*, k^*; v^*) < V^0(x^*, k^*)$. (This, of course, is simply a manifestation of the principle of optimality.) It follows that $v^0(x, k)$ may be defined as $C(x, V^0, k)$ and $V^0 \in \mathcal{V}^0$. These conditions together with the definition of T imply $V^0 = T(V^0)$.

Next we give necessary and sufficient conditions for optimality. The crucial step in the proof is due to Kalman-Caratheodory.

Theorem 1. $V = V^0$ IFF $V \in \mathcal{V}^0$ and $H^0(x, V, k) = 0$ in $G-S$

proof: Necessity follows directly from Lemmas 1(b) and 2(c). To show sufficiency, assume $H^0(x, V, k) = 0$ in $G-S$. Then for any $(x_0, k_0) \in G-S$

$$\begin{aligned} V^0(x_0, k_0) - V(x_0, k_0) &= \min_{v \in \mathcal{K}^0} [J(x_0, k_0; v) - V(x_0, k_0)] \\ &= \min_{v \in \mathcal{K}^0} \left[\lambda(x_1, k_1) + \sum_{\alpha=k_0}^{k_1-1} L(x, v, \alpha) - V(x_1, k_1) + \sum_{\alpha=k_0}^{k_1-1} \Delta V(x, \alpha) \right] \\ &= \min_{v \in \mathcal{K}^0} \left[\sum_{\alpha=k_0}^{k_1-1} H(x, V, \alpha, v) \right] \end{aligned}$$

Since $V \in \mathcal{V}^0$ choose $v(x, \alpha) = C(x, V, \alpha)$. Then $H(x, V, \alpha, v) = 0$ and

$$\min_{v \in K^0} \left[\sum_{\alpha=k_0}^{k_1-1} H(x, V, \alpha, v) \right] = 0 \text{ which implies that } V^0(x_0, k_0) - V(x_0, k_0) = 0.$$

A more familiar statement of this result is that, if the indicated minimization leads to an admissible control law, then a necessary and sufficient condition for optimality is

$$V(x, k) = \min_{v \in Q(x, k)} \left[L(x, u, k) + V(f(x, u, k), k+1) \right]$$

with $V(x, k) = \lambda(x, k)$ on S .

Corollary. $V = V^0$ IFF $T(V) = V$

Thus, T has one and only one fixed point, namely V^0 .

DEVELOPMENT OF THE BASIC INEQUALITIES

Lemma 3. Suppose $V \in \mathcal{V}^0$ and $W \in \mathcal{V}$. Then if

$$H^0(x, V, k) \leq H^0(x, W, k) \quad \text{in } G-S$$

it follows that $W(x, k) \leq V(x, k)$ in G

proof: Since $V \in \mathcal{V}^0$ the control $v(x, k) = C(x, V, k)$ is admissible. Taking forward differences along the motion of (1),

$$\begin{aligned} \Delta V - \Delta W &= V(f(x, v, k), k+1) - V(x, k) - W(f(x, v, k), k+1) + W(x, k) + L(x, v, k) - \\ &\quad L(x, v, k) \\ &= \left[H^0(x, V, k) - H^0(x, W, k) \right] + \left[H^0(x, W, k) - H(x, W, k, v) \right] \leq 0 \end{aligned}$$

since the term in the first bracket is nonpositive by assumption whereas the term in the second bracket is nonpositive by the definition of H^0 . It

follows that along $(\varphi_v(k), k)$, $\Delta V(x, k) \leq \Delta W(x, k)$. Note that $W(x_1, k_1) = V(x_1, k_1)$. Therefore

$$\Delta V \leq \Delta W \Rightarrow \sum_{\alpha=k_0}^{k_1-1} \Delta V \leq \sum_{\alpha=k_0}^{k_1-1} \Delta W \Rightarrow V(x_1, k_1) - V(x_0, k_0) \leq W(x_1, k_1) - W(x_0, k_0)$$

$$\Rightarrow W(x_0, k_0) \leq V(x_0, k_0) \quad \text{for all } (x_0, k_0) \in G-S$$

Theorem 2. Let $V \in \mathcal{V}^0$ and $W \in \mathcal{V}$. Then

$$H^0(x, V, k) \leq 0 \Rightarrow V^0(x, k) \leq V(x, k) \quad \text{in } G$$

$$H^0(x, W, k) \geq 0 \Rightarrow W(x, k) \leq V^0(x, k) \quad \text{in } G$$

proof: The conclusion follows from Lemma 3 and the fact that $V^0 \in \mathcal{V}^0$, $H^0(x, V^0, k) = 0$.

Note that the above results also hold for strict inequalities.

Theorem 2 provides a basic means of determining upper and lower bounds on $V^0(x, k)$.

Recall that Lemma 1 gives methods for determining $W \in \mathcal{V}$ such that $H(x, W, k, v) = 0$ and thus $H^0(x, W, k) \leq 0$. If it turns out that $W \in \mathcal{V}^0$ the function takes on a usefulness which is indicated in the following theorem.

Theorem 3. Let $W \in \mathcal{V}^0$ and $W^* = T(W)$. Then

$$H^0(x, W, k) \leq 0 \text{ in } G-S \Rightarrow V^0(x, k) \leq W^*(x, k) \leq W(x, k)$$

proof: Taking forward differences over the motion of (1) with $v(x, k) = C(x, W, k)$, we have

$$\Delta W = H^0(x, W, k) - L(x, v, k) \leq -L(x, v, k)$$

whereas from (11) $\Delta W^* = \Delta J(x, k; v) = -L(x, v, k)$. The desired result, $W^*(x, k) \leq W(x, k)$, follows by summation.

MONOTONE DECREASING SEQUENCES

Given any $V^1 \in \mathcal{V}^0$ we establish a successive approximation procedure through T , that is $V^{n+1} = T(V^n)$, by making the following assumption.

Assumption: $T(V^n) \in \mathcal{V}^0$ for $n = 1, 2, 3, \dots$

Theorem 4. If $V^1 \in \mathcal{V}^0$ and $V^{n+1} = T(V^n)$ $n = 1, 2, 3, \dots$ then

$$V^0(x, k) \leq V^{n+1}(x, k) \leq V^n(x, k) \leq V^1(x, k) \quad \text{in } G$$

proof: Let $V^{n+1}(x, k) = C(x, V^n, k) = T_1(V^n)$. Note that $V^n(x, k) = T_2(V^n(x, k)) = J(x, k; v^n)$ by definition. Therefore, by Lemma 1 we have, for each n

$$H(x, V^n, k, v^n) = 0$$

Thus $H^0(x, V^n, k) \leq 0$ and the conclusion follows directly from Theorem 3.

We may state stronger results if further conditions are met.

Theorem 5. If the iteration procedure terminates in a finite number of steps, that is, if $T(V^n) = V^n$ for some finite n , then $V^0(x, k) = V^n(x, k)$ on G .

proof: Since V^n is a fixed point in $\mathcal{V}^0$, the result follows directly from the Corollary to Theorem 1.

If the iteration does not end in a finite number of steps, Lemma 4 follows from Theorem 4.

Lemma 4. For each sequence $\{V^n\}$, there exists a function V^* such that $V^n(x, k) \downarrow V^*(x, k)$ pointwise on G .

As in [1], further assumptions on T , V^* can be made to yield stronger convergence conditions.

MONOTONE INCREASING SEQUENCES

Bellman [3] first showed that monotone increasing sequences converging to the optimal value function exist for certain classes of optimization problems. In the following we prove monotone convergence for a large class of nonlinear, stationary control systems. We make the following assumptions.

- Assumptions:
1. $G = \{(x, k) \mid k \leq k_1\}$
 2. $S = \{(x, k) \mid k = k_1\}$
 3. It follows from 1 and 2 that $\Omega(x, k) = U$ for $k < k_1$
 4. $\lambda = 0$
 5. $L(x, u, k) = L(x, u) \geq 0$ for all $x, u \in U$
 6. $f(x, u, k) = f(x, u)$

Theorem 6. For any $k < k_1 - 1$, and any x

$$V^0(x, k) \geq V^0(x, k+1) \geq \dots \geq V^0(x, k_1) = 0 \quad (16)$$

proof: We note that by Theorem 1

$$V^0(x, k) = \min_{u \in U} [L(x, u) + V^0(f(x, u), k+1)] \quad , \quad V^0(x, k_1) = 0$$

We use induction. In the first step

$$V^0(x, k_1 - 1) = \min_{u \in U} [L(x, u) + 0] \geq 0 = V^0(x, k_1)$$

Assuming $V^0(x, k) \geq V^0(x, k+1)$ we now show that $V^0(x, k-1) \geq V^0(x, k)$.
Thus

$$\begin{aligned} V^0(x, k) &\geq V^0(x, k+1) \Rightarrow V^0(f(x, u), k) \geq V^0(f(x, u), k+1) \\ &\Rightarrow V^0(f(x, u), k) + L(x, u) \geq V^0(f(x, u), k+1) + L(x, u) \\ \Rightarrow \min_{u \in U} [V^0(f(x, u), k) + L(x, u)] &\geq \min_{u \in U} [V^0(f(x, u), k+1) + L(x, u)] \\ &\Rightarrow V^0(x, k-1) \geq V^0(x, k) \end{aligned}$$

Note that we can consider the above monotone sequences as successive approximations to the solution of an N-stage decision process ($N = k_1 - k_0$) and write (16) in the more familiar notation.

$$V_N(x) \geq V_{N-1}(x) \geq \dots \geq V_0(x) = 0$$

The following simple example illustrates the use of the results.

Example:

$$x(k+1) = x(k) + u(k), \quad k = 0, 1, \dots, \quad u, x \in \mathbb{R}$$

$$J(x; u) = \sum_{k=0}^{\infty} x^2(k) + u^2(k)$$

monotone decreasing sequences

$v^1(x) = -x$ is an admissible control law. By Lemma 1.

$$V^1(x) = 2x^2$$

$$H(x, V^1, u) = 2(x+u)^2 - x^2 + u^2$$

and

$$C(x, V^1) = -\frac{2}{3}x$$

Therefore $V^1 \in \mathcal{V}^0$. It is clear that $T(V^n) \in \mathcal{V}^0$, $n = 1, 2, \dots$ in this case so that Theorem 4 holds. Setting $v^2(x) = -\frac{2}{3}x$ and repeating the above process yields the following successive approximations

$$V^1(x) = 2x^2$$

$$v^1(x) = -x$$

$$V^2(x) = 1.625x^2$$

$$v^2(x) = -.66x$$

$$V^3(x) = 1.618x^2$$

$$v^3(x) = -.62x$$

monotone increasing sequences

$$V_0(x) = 0$$

$$V_1(x) = \min_u [x^2 + u^2] = x^2$$

$$v_1(x) = 0$$

$$V_2(x) = \min_u [x^2 + u^2 + (x+u)^2] = 1.555x^2$$

$$v_2(x) = -.66x$$

$$V_3(x) = \min_u [x^2 + u^2 + \frac{14}{9}(x+u)^2] = 1.608x^2$$

$$v_3(x) = -.61x$$

so $1.608x^2 \leq V^0(x) \leq 1.618x^2$

REFERENCES

- [1] Leake, R. J. and Liu, Ruey-Wen, "Construction of Suboptimal Control Sequences," J. SIAM Control, Vol. 5, No. 1, 1967, 54-63.
- [2] Kalman, R. E., "The Theory of Optimal Control and the Calculus of Variations," Mathematical Optimization Techniques, R. Bellman, ed., University of California Press, Berkeley, 1963, Chap. 16.
- [3] Bellman, R. E., Dynamic Programming, Princeton University Press, Princeton, N. J., 1957.

(This work was supported in part by the Office of Naval Research, Project Themis, Contract N0014-68-A-0152)

A MAXIMUM PRINCIPLE FOR DISCRETE SYSTEMS WITH CONTROL VARIABLE INEQUALITY CONSTRAINTS

JUERGEN BRUCKNER and SHERMAN WU
Department of Electrical Engineering
Marquette University
Milwaukee, Wisconsin

ABSTRACT

A maximum principle is developed for discrete control systems in which the admissible control region is a function of the systems present state. A "Hamiltonian" is formulated and a set of necessary conditions for an optimal trajectory are then derived. A second order example is presented to illustrate the dependency of the optimal control on the allowable region.

INTRODUCTION AND PROBLEM FORMULATION

In general, optimization of discrete systems has only been studied with the restriction that the admissible controls belong to a fixed subset U of a Euclidean space E^m , $1, 2, 3$. In this paper, the admissible controls, $u(j)$, will belong to a subset $U \subset E^m$ which depends on the state of the system $x(j) \in E^n$. The evolution variable will assume the discrete values $0, 1, \dots, k$. The evolution of the system will be given by the vector difference equation

$$x(i+1) - x(i) = f[x(i), u(i), i] \quad i=0, 1, \dots, k-1 \quad (1)$$

where the control $u(i)$ is admissible, i.e.,

$$u(i) \in U[x(i)] \quad i=0, 1, \dots, k-1 \quad (2)$$

In the present paper let $T_i = \{0, 1, \dots, j\}$. For every $i \in T_{k-1}$, the vector valued function $f[x, u, i]$ is given and satisfies the following conditions:

1. the function $f[x, u, i]$ is defined and continuous for all $(x, u) \in E^n \times U[x(i)]$,
2. for every $(x, u) \in E^n \times U$, the function $f[x, u, i]$ is twice continuously differentiable with respect to x and u ,
3. the function $f[x, u, i]$ and all its first and second partials with respect to x and u are uniformly bounded over $A \times U$ for any bounded set $A \subset E^n$ and uniformly bounded over $E^n \times B$ for any bounded set $B \subset U \subset E^m$,
4. the matrix $I + \partial f / \partial x$ is not singular on $E^n \times U$, and
5. the set $\{f[x(i), u(i), i] : u(i) \in U[x(i)]\}$ is convex for every $x \in E^n$.

Conditions 1., 4., and 5. are essentially the same as those required in Halkin's paper [2], while conditions 2. and 3. have been strengthened due to the relaxation of the admissible control region.

Define an initial set

$$\{x: h_i(x) = 0 \quad i=1, 2, \dots, h\} \quad (3)$$

a terminal set

$$\{\underline{x}; g_1(\underline{x}) = 0 \quad i=1,2,\dots,j\} \quad (4)$$

and an objective function $g_0(\underline{x})$. The functions $h(\underline{x})$ and $g(\underline{x})$ are twice continuously differentiable in E^h and E^{j+1} respectively.

The subset $U[\underline{x}]$ is defined by

$$\begin{aligned} \sigma_\alpha[\underline{x}, \underline{u}] &= 0 \quad \alpha=1,2,\dots,l \\ \pi_\beta[\underline{x}, \underline{u}] &\leq 0 \quad \beta=1,2,\dots,s \end{aligned} \quad (5)$$

where σ_α and π_β as well as their first and second partials are defined and continuous on $E^n \times E^m$.

Two sequences, $\hat{\underline{u}}(0), \hat{\underline{u}}(1), \dots, \hat{\underline{u}}(k-1)$, and the corresponding trajectory, $\hat{\underline{x}}(0), \hat{\underline{x}}(1), \dots, \hat{\underline{x}}(k)$, are said to be optimal if they satisfy the conditions:

1. $h_1[\underline{x}(0)] = 0, \quad i=1,2,\dots,h$ (6)
2. $\underline{x}(i+1) - \underline{x}(i) = f[\underline{x}(i), \underline{u}(i), i] \quad \forall i \in T_{k-1}$ (7)
3. $\underline{u}(i) \in U[\underline{x}(i)] \quad \forall i \in T_{k-1}$ (8)
4. $g_1[\underline{x}(k)] = 0, \quad i=1,2,\dots,j$ and (9)
5. if $g_0[\hat{\underline{x}}(k)]$ is the maximum value of $g_0[\underline{x}(k)]$ subject to these constraints. (10)

SOLUTION

If the sequences $\hat{\underline{u}}(0), \hat{\underline{u}}(1), \dots, \hat{\underline{u}}(k-1), \hat{\underline{x}}(0), \hat{\underline{x}}(1), \dots, \hat{\underline{x}}(k)$ are optimal, then there exists a sequence of nonzero vectors, $\hat{\underline{p}}(0), \hat{\underline{p}}(1), \dots, \hat{\underline{p}}(k)$ such that

1. Admissible controls

a) $\underline{R}[\underline{x}, \underline{u}] = \underline{0}$ where $\underline{R}$ is a vector valued function signifying

$$\begin{aligned} \sigma_\beta[\underline{x}(i), \underline{u}(i)] &= 0 \quad \beta=1,2,\dots,l \quad \forall i \in T_{k-1} \\ \pi_\alpha[\underline{x}(i), \underline{u}(i)] &= 0 \quad \alpha=1,2,\dots,r \leq s \quad \forall i \in T_{k-1} \end{aligned}$$

and

b) $\pi_\alpha[\underline{x}(i), \underline{u}(i)] < 0 \quad \alpha=r+1,\dots,s \quad \forall i \in T_{k-1}$ (11)

2. Maximization of the Hamiltonian

$$f[\underline{x}(i), \underline{u}(i), i] \cdot \hat{\underline{p}}(i+1) \geq f[\hat{\underline{x}}(i), \underline{u}(i), i] \cdot \hat{\underline{p}}(i+1) \quad \forall i \in T_{k-1} \text{ and } \underline{u}(i) \in U[\underline{x}(i)] \quad (12)$$

3. Adjoint equations

$$\hat{\underline{p}}(i) - \hat{\underline{p}}(i+1) = \left\{ \frac{\partial f}{\partial \underline{x}} - \frac{\partial f}{\partial \underline{u}} \left(\frac{\partial \underline{R}}{\partial \underline{u}} \right)^{-1} \frac{\partial \underline{R}}{\partial \underline{x}} \right\}^T \begin{vmatrix} \hat{\underline{p}}(i+1) \\ \hat{\underline{x}} = \hat{\underline{x}}(i) \\ \hat{\underline{u}} = \hat{\underline{u}}(i) \end{vmatrix} \quad (13)$$

4. Transversality conditions

$$a) \hat{\underline{p}}(0) = \sum_{i=1}^h \alpha_i \left[\frac{\partial}{\partial \underline{x}} h_i(\underline{x}) \right]^T \Big|_{\underline{x}=\hat{\underline{x}}(0)} \quad (14)$$

$$b) \hat{\underline{p}}(k) = \sum_{i=0}^j \beta_i \left[\frac{\partial}{\partial \underline{x}} g_i(\underline{x}) \right]^T \Big|_{\underline{x}=\hat{\underline{x}}(k)} \quad (15)$$

$$c) \beta_0 \geq 0 \quad (16)$$

PROOF OF THE MAXIMUM PRINCIPLE

Define a set W of all states $\underline{x}(k)$ corresponding to all admissible sequences. Next define $S[\underline{x}(k)]$ as the set of all states satisfying Eq. (9) and for which the objective function takes on a value greater than at $\hat{\underline{x}}(k)$. Formally

$$S[\hat{x}(k)] = \{x: g_1[x] = 0 \quad i=1,2,\dots,j, \text{ and } g_0[x] > g_0[\hat{x}(k)]\} \quad (17)$$

The sets W and $S[\hat{x}(k)]$ are disjoint. If $f[x, u, i]$ is linear in state, the sets W and $S[\hat{x}(k)]$ are convex and hence separated. For a nonlinear problem, the sets are not necessarily convex and hence not necessarily separated. Halkin used the technique of relaxing a problem [4], to arrive at his necessary conditions for optimality. He considered a linearized problem about the optimal solution defined as follows:

1. the functions $h_1(x)$ are replaced by the functions

$$h_1[\hat{x}(0)] + [\partial h_1 / \partial x] \Big|_{x=\hat{x}(0)} [x - \hat{x}(0)] \quad (18)$$

2. the functions $g_1(x)$ are replaced by the functions

$$g_1[\hat{x}(k)] + [\partial g_1 / \partial x] \Big|_{x=\hat{x}(k)} [x - \hat{x}(k)] \quad (19)$$

3. the functions $f_j[x, u, i]$ are replaced by the functions

$$f_j[\hat{x}(i), \hat{u}(i), i] + [\partial f_j / \partial x, \partial f_j / \partial u] \Big|_{x=\hat{x}(i), u=\hat{u}(i)} [x - \hat{x}(i), u - \hat{u}(i)] \quad (20)$$

Next he defined W_H^+ and $S_H^+[\hat{x}(k)]$ as before but with respect to the linearized problem above. Halkin then proved after great lengths, that if the control region is constant and if the sets W and $S[\hat{x}(k)]$ are disjoint, then the sets W_H^+ and $S_H^+[\hat{x}(k)]$ are separated ("The Linearization Lemma").

In this paper the control region is not constant. A linearized problem about the optimal solution is again considered but with the functions $f_j[x(i), u(i), i]$ being replaced by the functions

$$f_j[\hat{x}(i), \hat{u}(i), i] + [\partial f_j / \partial x, \partial f_j / \partial u] \Big|_{x=\hat{x}(i), u=\hat{u}(i)} [x - \hat{x}(i), u - \hat{u}(i)] \quad (21)$$

Now define $S^+[\hat{x}(k)]$ and W^+ with respect to this linearization. Eq. (21) must be reduced to some form of Eq. (20) if Halkin's results are to be used.

Suppose that the optimal control $\hat{u}(i)$ and the corresponding trajectory $\hat{x}(i)$ are such that $r \leq s$ of the constraints in Eq. (5) are equalities, that is, renumbering if necessary,

$$\begin{aligned} \pi_\beta[\hat{x}(i), \hat{u}(i)] &= 0 \quad \beta=1,2,\dots,r \leq s \quad \forall i \in T_{k-1} \\ \pi_\beta[\hat{x}(i), \hat{u}(i)] &< 0 \quad \beta=r+1,\dots,s \end{aligned} \quad (22)$$

Assume the matrix

$$\begin{bmatrix} \partial \sigma_1 / \partial u_1 & \dots & \partial \sigma_1 / \partial u_m \\ \partial \sigma_1 / \partial u_1 & \dots & \partial \sigma_1 / \partial u_m \\ \partial \pi_1 / \partial u_1 & \dots & \partial \pi_1 / \partial u_m \\ \partial \pi_r / \partial u_1 & \dots & \partial \pi_r / \partial u_m \end{bmatrix}$$

has maximum rank at $\underline{x} = \hat{\underline{x}}(i)$, $\underline{u} = \hat{\underline{u}}(i)$, for all $i \in T_{k-1}$. Then there is a neighborhood of $(\underline{x}, \underline{u})$ in $E^n \times E^m$ such that

$$\begin{aligned} \alpha_\alpha(\underline{x}, \underline{u}) &= 0 \quad \alpha=1, 2, \dots, l \\ \eta_\beta(\underline{x}, \underline{u}) &= 0 \quad \beta=1, 2, \dots, r \end{aligned} \quad (23)$$

may be solved uniquely for $r+1$ components of $\underline{u}$ as functions of $\underline{x}$ and the remaining $m - (r+1)$ components. Let

$$\begin{aligned} \underline{\lambda} &= (u_1, u_2, \dots, u_{r+1})^T \\ \underline{\theta} &= (u_{r+1+1}, \dots, u_m)^T \end{aligned} \quad (24)$$

and write Eq. (23) as

$$R(\underline{x}, \underline{\lambda}, \underline{\theta}) = 0 \quad (25)$$

Then, since $R(\underline{x}, \underline{\lambda}, \underline{\theta})$ has its first and second partials defined and continuous, there exists a function $V(\underline{x}, \underline{\theta})$ such that

$$R[\underline{x}, \underline{\theta}, V(\underline{x}, \underline{\theta})] = 0 \quad (26)$$

whose first and second partials are also defined and continuous in a neighborhood of $(\underline{x}, \underline{u})$ in $E^n \times E^m$. Let

$$\theta_j(i) = \hat{u}_j(i) \quad j=r+1+1, \dots, m \quad (27)$$

Then

$$\underline{u}(i) = [\underline{\lambda}(i), \underline{\theta}(i)]^T \quad (28)$$

and the linearized system equation becomes

$$\begin{aligned} \underline{x}(i+1) - \underline{x}(i) &= f[\hat{\underline{x}}(i), \hat{\underline{u}}(i), i] \\ &+ [\partial f[\underline{x}, \hat{\underline{u}}(i), i] / \partial \underline{x}] \left[\begin{array}{c} \underline{x}(i) - \hat{\underline{x}}(i) \\ \underline{x} = \hat{\underline{x}}(i) \end{array} \right] \\ &+ [\partial f[\hat{\underline{x}}(i), \underline{u}, i] / \partial \underline{\lambda}] \left[\begin{array}{c} \underline{\lambda}(i) - \hat{\underline{\lambda}}(i) \\ \underline{u} = \hat{\underline{u}}(i) \end{array} \right] \end{aligned} \quad (29)$$

On the linearized trajectory

$$\underline{\lambda}(i) - \hat{\underline{\lambda}}(i) = [\partial V / \partial \underline{\lambda}] \left[\begin{array}{c} \underline{x}(i) - \hat{\underline{x}}(i) \\ \underline{x} = \hat{\underline{x}}(i) \end{array} \right] \quad (30)$$

and

$$[\partial R / \partial \underline{x}][\underline{x}(i) - \hat{\underline{x}}(i)] + [\partial R / \partial \underline{\lambda}][\underline{\lambda}(i) - \hat{\underline{\lambda}}(i)] = 0 \quad (31)$$

where $\partial R / \partial \underline{x}$ and $\partial R / \partial \underline{\lambda}$ are evaluated at $[\hat{\underline{x}}(i), \hat{\underline{u}}(i)]$.

By assumption $\partial R / \partial \underline{\lambda}$ has maximum rank. Hence

$$\underline{\lambda}(i) - \hat{\underline{\lambda}}(i) = -[\partial R / \partial \underline{\lambda}]^{-1} [\partial R / \partial \underline{x}][\underline{x}(i) - \hat{\underline{x}}(i)] \quad (32)$$

$\underline{\lambda}(i)$ is admissible since Eq. (23) is satisfied and for sufficiently small $\|\underline{x}(i) - \hat{\underline{x}}(i)\|$,

$$\begin{aligned}
\pi_{\beta} [\underline{x}(i), \underline{u}(i)] &= \pi_{\beta} [\hat{\underline{x}}(i), \hat{\underline{u}}(i)] \\
&+ \left\{ \partial \pi_{\beta} / \partial \lambda [\underline{a} \underline{v} / \partial \underline{x}] + \partial \pi_{\beta} / \partial \underline{x} \right\} \left[\begin{array}{l} \underline{x}(i) - \hat{\underline{x}}(i) \\ \underline{x} = \hat{\underline{x}}(i) \\ \underline{u} = \hat{\underline{u}}(i) \end{array} \right] \\
&+ o \left\| \underline{x}(i) - \hat{\underline{x}}(i) \right\| \\
&< 0 \quad \beta = r+1, \dots, s
\end{aligned}$$

since

$$\pi_{\beta} [\hat{\underline{x}}(i), \hat{\underline{u}}(i)] < 0 \quad \beta = r+1, \dots, s$$

Eq. (21) may then be written

$$\begin{aligned}
&\underline{f}[\hat{\underline{x}}(i), \hat{\underline{u}}(i), i] + \left\{ \partial \underline{f}[\underline{x}, \hat{\underline{u}}(i), i] / \partial \underline{x} \right\} \Big|_{\substack{\underline{x} = \hat{\underline{x}}(i) \\ \underline{u} = \hat{\underline{u}}(i)}} \left[\begin{array}{l} \underline{x}(i) - \hat{\underline{x}}(i) \\ \underline{x} = \hat{\underline{x}}(i) \end{array} \right] \\
&+ \left[\partial \underline{f}[\hat{\underline{x}}(i), \underline{u}, i] / \partial \lambda \right] \Big|_{\substack{\underline{x} = \hat{\underline{x}}(i) \\ \underline{u} = \hat{\underline{u}}(i)}} \left[\begin{array}{l} \partial \underline{v}[\underline{x}, \underline{u}(i)] / \partial \underline{x} \\ \underline{x} = \hat{\underline{x}}(i) \end{array} \right] \left[\begin{array}{l} \underline{x}(i) - \hat{\underline{x}}(i) \\ \underline{x} = \hat{\underline{x}}(i) \end{array} \right] \quad (33)
\end{aligned}$$

By requiring $\underline{f}$ to be twice continuously differentiable with respect to $\underline{x}$ and $\underline{u}$, and requiring $\underline{R}[\underline{x}, \underline{u}]$ to be twice continuously differentiable with respect to $\underline{x}$, Halkin's requirements on his linearized equations are met and the Linearization Lemma can be used.

Substituting Eq. (22) into Eq. (29), the variational equation can be written

$$\begin{aligned}
&[\underline{x}(i+1) - \hat{\underline{x}}(i+1)] - [\underline{x}(i) - \hat{\underline{x}}(i)] \\
&= \left\{ \frac{\partial}{\partial \underline{x}} \underline{f}[\underline{x}, \hat{\underline{u}}(i), i] \right. \\
&\quad \left. - \frac{\partial}{\partial \underline{u}} \underline{f}[\hat{\underline{x}}(i), \underline{u}, i] \left[\frac{\partial \underline{R}}{\partial \underline{u}} \right]^{-1} \left[\frac{\partial \underline{R}}{\partial \underline{x}} \right] \right\} \left[\begin{array}{l} \underline{x}(i) - \hat{\underline{x}}(i) \\ \underline{x} = \hat{\underline{x}}(i) \\ \underline{u} = \hat{\underline{u}}(i) \end{array} \right] \quad (34)
\end{aligned}$$

from which the adjoint equation is seen to be

$$\begin{aligned}
&\hat{\underline{p}}(i) - \hat{\underline{p}}(i+1) = \hat{\underline{p}}(i+1) \left\{ \frac{\partial}{\partial \underline{x}} \underline{f}[\underline{x}, \hat{\underline{u}}(i), i] \right. \\
&\quad \left. - \frac{\partial}{\partial \underline{u}} \underline{f}[\hat{\underline{x}}(i), \underline{u}, i] \left[\frac{\partial \underline{R}}{\partial \underline{u}} \right]^{-1} \left[\frac{\partial \underline{R}}{\partial \underline{x}} \right] \right\} \Big|_{\substack{\underline{x} = \hat{\underline{x}}(i) \\ \underline{u} = \hat{\underline{u}}(i)}} \quad (35)
\end{aligned}$$

or rewriting the above using column vectors

$$\hat{\underline{p}}(i) - \hat{\underline{p}}(i+1) = \left\{ \frac{\partial \underline{f}}{\partial \underline{x}} - \frac{\partial \underline{f}}{\partial \underline{u}} \left[\frac{\partial \underline{R}}{\partial \underline{u}} \right]^{-1} \frac{\partial \underline{R}}{\partial \underline{x}} \right\}^T \left[\begin{array}{l} \hat{\underline{p}}(i+1) \\ \underline{x} = \hat{\underline{x}}(i) \\ \underline{u} = \hat{\underline{u}}(i) \end{array} \right] \quad (36)$$

Since the Linearization Lemma applies, the sets W^+ and $S^+[\hat{\underline{x}}(k)]$ are convex and separated, i.e., there exists a non-zero vector $\hat{\underline{d}}$ such that

$$[\underline{x} - \hat{\underline{x}}(k)] \cdot \hat{\underline{d}} \geq 0 \quad \forall \underline{x} \in S^+[\hat{\underline{x}}(k)] \quad (37)$$

$$[\underline{x} - \hat{\underline{x}}(k)] \cdot \hat{\underline{d}} \leq 0 \quad \forall \underline{x} \in W^+ \quad (38)$$

The sequence of nonzero vectors $\hat{p}(0), \hat{p}(1), \dots, \hat{p}(k)$ is defined as the solution of the difference equation (35) with the terminal condition

$$\hat{p}(k) = \delta \quad (39)$$

For every $i \in T_{k-1}$

$$[\tilde{x}(i) - \hat{x}(i)] \cdot \hat{p}(i) = [\tilde{x}(i+1) - \hat{x}(i+1)] \cdot \hat{p}(i+1) \quad (40)$$

Proof: From Eq. (32),

$$-\hat{p}^T(i+1) \left[\frac{\partial f}{\partial \lambda} \right] \left\{ \left[\frac{\partial R}{\partial \lambda} \right]^{-1} \left[\frac{\partial R}{\partial x} \right] [\tilde{x}(i) - \hat{x}(i)] + [\tilde{\lambda}(i) - \hat{\lambda}(i)] \right\} = 0 \quad (41)$$

But from Eq. (36)

$$-\hat{p}^T(i+1) \left[\frac{\partial f}{\partial \lambda} \right] \left[\frac{\partial R}{\partial \lambda} \right]^{-1} \left[\frac{\partial R}{\partial x} \right] = \hat{p}^T(i) - \hat{p}^T(i+1) - \hat{p}^T(i+1) \left[\frac{\partial f}{\partial x} \right]$$

and Eq. (41) becomes

$$\left\{ \hat{p}^T(i) - \hat{p}^T(i+1) - \hat{p}^T(i+1) \left[\frac{\partial f}{\partial x} \right] \right\} [\tilde{x}(i) - \hat{x}(i)] - \hat{p}^T(i+1) \left[\frac{\partial f}{\partial \lambda} \right] [\tilde{\lambda}(i) - \hat{\lambda}(i)] = 0 \quad (42)$$

$$-\hat{p}^T(i+1) \left\{ [\tilde{x}(i) - \hat{x}(i)] + \left[\frac{\partial f}{\partial x} \right] [\tilde{x}(i) - \hat{x}(i)] + \left[\frac{\partial f}{\partial \lambda} \right] [\tilde{\lambda}(i) - \hat{\lambda}(i)] \right\} + \hat{p}^T(i) [\tilde{x}(i) - \hat{x}(i)] = 0 \quad (43)$$

Substituting Eq. (32) and Eq. (34) into Eq. (43) yields

$$-\hat{p}^T(i+1) [\tilde{x}(i+1) - \hat{x}(i+1)] + \hat{p}^T(i) [\tilde{x}(i) - \hat{x}(i)] = 0 \quad (44)$$

or

$$[\tilde{x}(i) - \hat{x}(i)] \cdot \hat{p}(i) = [\tilde{x}(i+1) - \hat{x}(i+1)] \cdot \hat{p}(i+1) \quad (45)$$

Now for any $i \in T_k$ let $W^+(i)$ be the set of all states reachable at the evolution time i for the linearized system (21) from all initial states defined by (18) and with all admissible control sequences. Then $W^+(k) = W^+[\hat{x}(k)]$. For every $i \in T_k$ and all $\underline{x} \in W^+(i)$

$$[\underline{x} - \hat{x}(i)] \cdot \hat{p}(i) \leq 0 \quad (46)$$

Proof: Assume contrary to the desired conclusion that for some $j \in T_k$, $\tilde{x}(j) \in W^+(j)$ and

$$[\tilde{x}(j) - \hat{x}(j)] \cdot \hat{p}(j) = \gamma > 0 \quad (47)$$

But

$$[\tilde{x}(i) - \hat{x}(i)] \cdot \hat{p}(i) = [\tilde{x}(i+1) - \hat{x}(i+1)] \cdot \hat{p}(i+1) \quad (48)$$

which then implies

$$[\tilde{x}(k) - \hat{x}(k)] \cdot \hat{p}(k) = \gamma > 0 \quad (49)$$

which contradicts Eq. (38) since $\hat{p}(k) = \delta$ and $\tilde{x}(k) \in W^+[\hat{x}(k)]$.

Eq. (12) will now be proven by contradiction. Assume that there exists a $j \in T_{k-1}$, a $\gamma > 0$, and a $\tilde{u}(j) \in U[\underline{x}(j)]$ such that

$$f[\hat{x}(j), \tilde{u}(j), j] \cdot \hat{p}(j+1) = f[\underline{x}(j), \hat{u}(j), j] \cdot \hat{p}(j+1) + \gamma \quad (50)$$

Now

$$\hat{x}(j+1) - \hat{x}(j) = f[\hat{x}(j), \hat{u}(j), j]$$

and

$$\tilde{x}(j+1) - \hat{x}(j) = f[\hat{x}(j), \tilde{u}(j), j]$$

so that

$$\begin{aligned} [\tilde{x}(j+1) - \hat{x}(j+1)] \cdot \hat{p}(j+1) &= [\hat{x}(j) + f[\hat{x}(j), \tilde{u}(j), j] \\ &\quad - [\hat{x}(j) + f[\hat{x}(j), \hat{u}(j), j]]] \cdot \hat{p}(j+1) \end{aligned} \quad (51)$$

$$\begin{aligned} &= f[\hat{x}(j), \tilde{u}(j), j] \cdot \hat{p}(j+1) \\ &\quad - f[\hat{x}(j), \hat{u}(j), j] \cdot \hat{p}(j+1) \\ &= \gamma > 0 \end{aligned} \quad (52)$$

contradicting Eq. (46).

The transversality conditions will next be proven. For all $\underline{x} \in W^+(0)$, Eq. (46) yields

$$[\underline{x} - \hat{x}(0)] \cdot \hat{p}(0) \leq 0 \quad (53)$$

and Eq. (18) yields

$$[\underline{x} - \hat{x}(0)] \cdot \left[\frac{\partial}{\partial \underline{x}} h_1(\underline{x}) \right]_{\underline{x}=\hat{x}(0)}^T = 0 \quad i=1, 2, \dots, h \quad (54)$$

Hence

$$[\underline{x} - \hat{x}(0)] \cdot \hat{p}(0) = 0 \quad (55)$$

for all $\underline{x}$ satisfying Eq. (54) and Eq. (14) is satisfied.

Similarly for all $\underline{x} \in S^+[\hat{x}(k)]$

$$[\underline{x} - \hat{x}(k)] \cdot \hat{p}(k) \geq 0 \quad (56)$$

that is for all $\underline{x}$ such that

$$[\underline{x} - \hat{x}(k)] \cdot \left[\frac{\partial}{\partial \underline{x}} g_1(\underline{x}) \right]_{\underline{x}=\hat{x}(k)}^T = 0 \quad i=1, 2, \dots, j \quad (57)$$

and

$$[\underline{x} - \hat{x}(k)] \cdot \left[\frac{\partial}{\partial \underline{x}} g_0(\underline{x}) \right]_{\underline{x}=\hat{x}(k)}^T > 0 \quad (58)$$

Hence

$$[\underline{x} - \hat{x}(k)] \cdot \hat{p}(k) = 0$$

for all $\underline{x}$ such that

$$[\underline{x} - \hat{x}(k)] \cdot \left[\frac{\partial}{\partial \underline{x}} g_i(\underline{x}) \right]_{\underline{x}=\hat{x}(k)}^T = 0 \quad i=0, 1, \dots, j \quad (59)$$

Finally since for all $\underline{x} \in S^+[\hat{\underline{x}}(k)]$

$$[\underline{x} - \hat{\underline{x}}(k)] \cdot \hat{\underline{p}}(k) \geq 0 \quad (60)$$

i.e.,
$$[\underline{x} - \hat{\underline{x}}(k)] \cdot \left\{ \sum_{i=1}^j \beta_i \left[\frac{\partial g_i(\underline{x})}{\partial \underline{x}} \right] \right\}_{\underline{x}=\hat{\underline{x}}(k)}^T = 0 \quad (61)$$

and
$$[\underline{x} - \hat{\underline{x}}(k)] \cdot \left[\frac{\partial g_0(\underline{x})}{\partial \underline{x}} \right]_{\underline{x}=\hat{\underline{x}}(k)}^T > 0 \quad (62)$$

$\beta_0 \geq 0$ because $S^+[\hat{\underline{x}}(k)]$ is not empty.

EXAMPLE

Let

$$x_1(i+1) - x_1(i) = -\frac{1}{2}x_1(i) + u_1(i)$$

$$x_2(i+1) - x_2(i) = \frac{1}{4}x_1(i) - u_2(i)$$

subject to

$$|u_1(i)| \leq 1$$

$$|u_2(i)| \leq 1 - \frac{1}{2} \left(\frac{17}{16} \right)^2 \frac{x_1^2(i)}{1 + x_1^2(i)}$$

Conditions 1. to 5. on $f[\underline{x}, \underline{u}, \underline{i}]$ are obviously satisfied. Define an initial set; $x_1(0) = x_2(0) = 0$, and an objective function: $g_0(\underline{x}) = x_2(2)$. The subset $U(\underline{x})$ is defined by:

$$\pi_1(\underline{x}, \underline{u}) = u_1 - 1$$

$$\pi_2(\underline{x}, \underline{u}) = -u_1 - 1 = \pi_1 - 2u_1$$

$$\pi_3(\underline{x}, \underline{u}) = u_2 - \left[1 - \frac{1}{2} \left(\frac{17}{16} \right)^2 \frac{x_1^2(i)}{1 + x_1^2(i)} \right]$$

$$\pi_4(\underline{x}, \underline{u}) = \pi_2 - 2u_2$$

The transversality conditions become

$$\hat{\underline{p}}(0) = [\alpha_1, \alpha_2]^T; \hat{\underline{p}}(k) = \hat{\underline{p}}(2) = [0, \beta]^T \text{ with } \beta \geq 0.$$

π_1 and π_2 have no bearing on the matrix $\partial R / \partial \underline{x}$. π_3 and π_4 cannot be satisfied simultaneously. Define:

$$\begin{aligned} a &= -\frac{1}{2} \left(\frac{17}{16} \right)^2 \frac{x_1(i)}{1 + x_1^2(i)} \\ B(i) &= 1 \text{ if } u_2(i) = 1 + a \cdot x_1(i) \\ &= 0 \text{ if } |u_2(i)| < 1 + a \cdot x_1(i) \\ &= -1 \text{ if } -u_2(i) = 1 + a \cdot x_1(i) \end{aligned}$$

Then

$$\frac{\partial R}{\partial \underline{x}} = \left[\frac{-2a}{1 + x_1^2(i)}, 0 \right]$$

$$\frac{\partial R}{\partial \underline{u}} = [B(i)]$$

and

$$\left[\frac{\partial f}{\partial u} \left(\frac{\partial R}{\partial u} \right)^{-1} \frac{\partial R}{\partial x} \right]_{\substack{x=\hat{x}(i) \\ u=\hat{u}(i)}} = \begin{bmatrix} 0 & 0 \\ \frac{2aB(i)}{1 + \hat{x}_1(i)} & 0 \end{bmatrix}$$

The adjoint difference equation can be written

$$\hat{p}(i) = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} + \frac{2aB(i)}{1 + \hat{x}_1(i)} \\ 0 & 1 \end{bmatrix} \hat{p}(i+1)$$

Maximizing the Hamiltonian requires

$$\begin{aligned} \hat{u}_1(i) &= 1 \cdot \text{signum } \hat{p}_1(i+1) \\ \hat{u}_2(i) &= -[1 + a \hat{x}_1(i)] \cdot \text{signum } \hat{p}_2(i+1) \end{aligned}$$

where

$$\begin{aligned} \text{signum}(x) &= 1 \text{ if } x > 0 \\ &= -1 \text{ if } x < 0 \end{aligned}$$

$$|\text{signum}(x)| \leq 1 \text{ if } x = 0$$

Since $\hat{p}(k) = \hat{p} \neq 0$, $p_2(2) = \beta > 0$, and

$$\hat{u}_2(i) = - \left[1 - \frac{1}{2} \left(\frac{17}{16} \right)^2 \frac{\hat{x}_1^2(i)}{1 + \hat{x}_1(i)} \right] \quad \forall i \in T_{k-1}$$

$$B(i) = -1 \quad \forall i \in T_{k-1}$$

$$\hat{p}_1(1) = \left[\frac{1}{4} - \left(\frac{17}{16} \right)^2 \frac{\hat{x}_1(1)}{[1 + \hat{x}_1(1)]^2} \right] \beta$$

The optimal value of $x_1(1)$ can be determined by finding the optimal value of $x_2(2)$ with respect to $x_1(1)$ subject to $|x_1(1)| \leq 1$, resulting in $\hat{x}_1(1) = \frac{1}{4}$. Hence

$$\begin{aligned} \hat{p}_1(1) &= \left[\frac{1}{4} - \left(\frac{17}{16} \right)^2 \frac{(\frac{1}{4})}{[1 + (\frac{1}{4})]^2} \right] \beta \\ &= 0 \end{aligned}$$

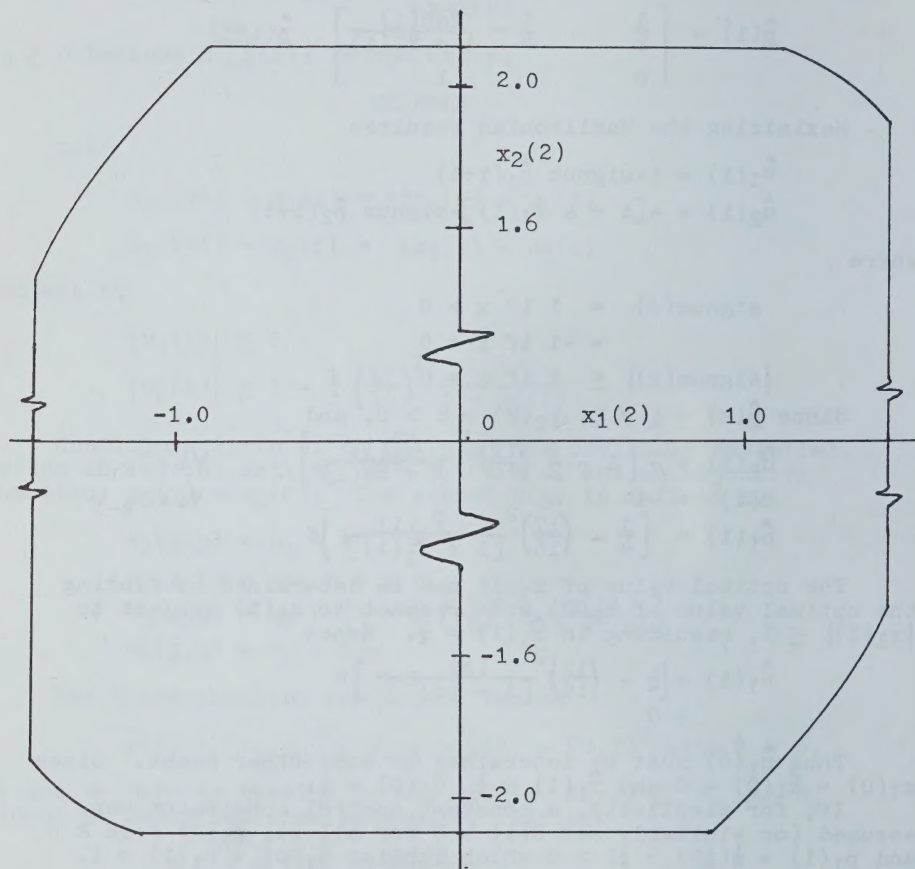
Thus $\hat{u}_1(0)$ must be determined by some other means. Since $x_1(0) = \hat{x}_1(0) = 0$ and $\hat{x}_1(1) = \frac{1}{4}$, $\hat{u}_1(0) = \frac{1}{4}$.

If, for simplicity, a constant control constraint was assumed (or similarly set $B(i) = 0$ for all i), $p_1(2) = \frac{1}{4}\beta > 0$ and $p_1(1) = \frac{1}{2}(\frac{1}{4}\beta) + \frac{1}{4}\beta > 0$ which implies $u_1(0) = u_1(1) = 1$. Then $\hat{x}_2(2) = 1 + 991/1024$ whereas $\hat{x}_2(2) = 2 + 15/128$.

REFERENCES

- [1] Butkovskii, A. G., "The Necessary and Sufficient Conditions For Optimality of Discrete Control Systems," Avtomatika i Telemekhanika, Vol. 24, No. 8, 1963. (English translation: Automation and Remote Control)
- [2] Halkin, H., "A Maximum Principle of the Pontryagin Type for Systems Described by Nonlinear Difference Equations," J. SIAM Control, Vol. 4, No. 1, 1966.

- [3] Holzman, J. M., "Convexity and the Maximum Principle for Discrete Systems," IEEE Transactions on Automatic Control, Vol. AC-11, No. 1, 1966.
- [4] Warga, J., "Relaxed Variational Problems," J. Math. Anal. and Appl., Vol. 4, 1962.



Reachable Set $W(2)$

Acknowledgement: This work was supported in part by the National Science Foundation under grant number GK-855.

IMPLEMENTING THE SOLUTION TO CONTINUOUS OPTIMAL CONTROL OF DISTRIBUTED PARAMETER SYSTEMS BY FINITE DIFFERENCES

GEORGE L. KUSIC
University of Pittsburgh
Pittsburgh, Pennsylvania 15213

ABSTRACT

Finite difference methods are used to implement the solution to optimal control of distributed parameter systems. The control is assumed to be intrinsic to the partial differential equation as well as continuous, such that calculus of variations is used to obtain the control law.

Several important principles are developed to formulate the difference approximations to the partial differential equations which describe the system and the control law. An iterative method of solution is employed on these two equations. The convergence of the iteration is assured by stability considerations of the finite difference expressions.

I. PROBLEM STATEMENT

Consider the distributed parameter system described by the non-constant coefficient, linear partial differential equation in two independent variables:

$$\frac{\partial^n u}{\partial t^n} = \sum_{i=0}^{n-1} \alpha'_i \frac{\partial^i u}{\partial t^i} + q(t, x) = \sum_{i=0}^{n-1} \sum_{\ell=0}^g f_{i\ell}(t, x) \frac{\partial^\ell}{\partial x^\ell} \frac{\partial^i u}{\partial x^i} + q(t, x) \quad (1)$$

where α'_i are non-constant coefficient operators such that $f_{i\ell}$ are continuous on the domain of the problem $R: \{0 \leq t \leq T, 0 \leq x \leq 1\}$, $q(t, x)$ is the control function. Given the n initial conditions $u(0, x)$,

$\frac{\partial u}{\partial t}(0, x) \dots \frac{\partial^{n-1} u}{\partial t^{n-1}}(0, x)$ and a total of g boundary conditions at $x = 0$ or

$x = 1$, it is desired to find the optimal control, $q(t, x)$ which minimizes the quadratic performance index:

$$J = \int_{x=0}^{x=1} \int_{t=0}^{x=T} \{ [u(t, x) - u_D(t, x)]^2 + \frac{q^2(t, x)}{W} \} \partial x \partial t \quad (2)$$

where $u_D(t, x)$ is a desired function and W is a relative weighting factor for the cost of control.

It is seen that this formulation is an isoperimetric problem in the calculus of variations^[1], with free terminal conditions, $u(T, x)$. This leads to a final value type of partial differential equation in either the Lagrange multiplier, e , or control, q , which are coupled to (1). In general this system of equations cannot be solved by a substitution of variables so that (1) and its adjoint equation must be solved simultaneously. Herein are presented several principles to solve this system of coupled partial differential equations by finite difference methods.

II. THE FINITE DIFFERENCE EQUATION

The domain of the problem is discretized by a uniform $M \times N$ mesh, $R_D: \{0 \leq m \Delta t \leq T, 0 \leq n \Delta x \leq 1\}$ where $m=0,1,2,\dots,M$ and $n=0,1,2,3,\dots,N$. The step sizes are accordingly given as $\Delta t = T/M$ and $\Delta x = 1/N$. Displacements on the mesh are used to approximate the space derivatives:

$$\frac{\partial^g u}{\partial x^g} = \sum_r \frac{\gamma_r u(t, x + r \Delta x)}{\Delta x^g} + O(\Delta x^h) \approx \sum_r \frac{\gamma_r u(t, x + r \Delta x)}{\Delta x^g} \quad (3)$$

where h is an integer which describes the degree of the approximation and there must be at least $g+1$ coefficients γ_r . In order for the right hand side of (3) to be a consistent approximation to $\frac{\partial^g u}{\partial x^g}$ as $\Delta x \rightarrow 0$, it is seen that the following must hold:

$$\sum_r \gamma_r = 0, \quad \sum_r \frac{r \gamma_r}{g} = 0, \quad \sum_r \frac{r^2 \gamma_r}{g(g-1)} = 0, \dots, \quad \sum_r \frac{r^g \gamma_r}{g!} = 1 \quad (4)$$

The Fourier transform of (3) yields:

$$F\left(\frac{\partial^g u}{\partial x^g}\right) \approx \sum_r \frac{\gamma_r e^{jkr \Delta x} u(t, k)}{\Delta x^g} \quad (5)$$

where k is the transform or dual variable. Note the right hand side of (5) can be separated into real and imaginary parts. The manner in which the real and imaginary parts approach zero is subsequently shown to be important in the stability of the difference approximation to the partial differential equation, hence the following definitions:

Definition 1: An even order derivative - 2, 4, 6... - is said to be approximated by a definite difference scheme when the imaginary part of the coefficients of its Fourier transform is uniquely zero and the real part is bounded from above or below by zero for all k . That is:

$$\text{Im}\{ \} = 0 \quad \text{Re}\{ \} \leq 0 \text{ or } \text{Re}\{ \} \geq 0 \quad (6)$$

Definition 2: An odd order derivative - 1, 3, 5... - is said to be approximated by a definite difference scheme when the real part of the coefficients of its Fourier transform are bounded from above or below by zero for all k , while for the same scheme the imaginary part may be expressed as:

$$\text{Im}\{ \} = \frac{\sin k \Delta x}{\Delta x^g} f(k \Delta x), \quad f(k \Delta x) \geq 0 \text{ or } f(k \Delta x) \leq 0 \quad (7)$$

A central difference scheme for even order derivatives yields the properties of definition 1 as seen by the example:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(t, x + \Delta x) - 2u(t, x) + u(t, x - \Delta x)}{\Delta x^2} + O(\Delta x) \quad (8)$$

Properties of the second definition are exhibited by:

$$\frac{\partial u}{\partial x} \approx \frac{u(t, x + \Delta x) - u(t, x)}{\Delta x} \quad (9a) \quad \frac{\partial u}{\partial x} \approx \frac{u(t, x) - u(t, x - \Delta x)}{\Delta x} \quad (9b)$$

This leads to the following lemma whose proof is by construction:

Lemma: An odd g^{th} derivative may be approximated by a $g+1$ point difference scheme which can have, as desired, either a positive or negative definite real part while the coefficient $f(k \Delta x)$ is definite.

The algebraic sign of $f(k\Delta x)$ is fixed, of course, for each particular g .

Temporarily consider (1) to be a constant coefficient partial difference equation ($f_{i,l}$ is a constant). When the space derivatives are approximated by difference schemes this constant coefficient equation has the following Fourier transform with respect to the space axis:

$$\frac{\partial^n u}{\partial t^n}(t, k) = \sum_{i=0}^{n-1} \sum_{l=0}^g f_{i,l} \frac{\gamma_r e^{jk\Delta x}}{(\Delta x)^l} \frac{\partial^i u}{\partial t^i}(t, k) + q(t, k)$$

$$= \sum_{i=0}^{n-1} \alpha_i \frac{\partial^i u}{\partial t^i} + q(t, k) \quad (10)$$

where α_i represents the Fourier coefficient of the difference approximation to $f_{i,l}$:

$$\alpha_i = \sum_l f_{i,l} \sum_r \frac{\gamma_r e^{jk\Delta x}}{(\Delta x)^l} \quad (11)$$

When the time derivative $\partial^n u / \partial t^n$ of (10) is approximated by a difference scheme of $n+1$ consecutive time steps (called the minimal polynomial approximation), the coefficients of these steps form a binomial expansion. Express $\partial^n u / \partial t^n$ by this approximation, and use a $p+1$ step consistent approximation for the lower derivatives $\partial^i u / \partial t^i$ where $i=0, 1, 2, 3, \dots, n-1$ and $p \geq n$.

Therefore there exists $p-i$ degrees of freedom from which to choose time derivative approximations for the i lower derivatives. This freedom is often essential to obtain a stable difference formulation.

The homogeneous part of (10) is considered here as the highest time derivative and the α_i which contain differentiation of the dependent variable ($l > 0$ in (10)). The characteristic polynomial of the homogeneous part is:

$$p_H(\lambda) = \lambda^p [1 - \Delta t \alpha_{n-1} b_0 - \Delta t^2 \alpha_{n-2} c_0 - \Delta t^3 \alpha_{n-3} d_0 \dots]$$

$$+ \lambda^{p-1} [-\Delta t \alpha_{n-1} b_1 - \Delta t^2 \alpha_{n-2} c_1 - \Delta t^3 \alpha_{n-3} d_1 \dots]$$

$$+ \lambda^{p-2} [\frac{n(n-1)}{2!} - \Delta t \alpha_{n-1} b_2 - \Delta t^2 \alpha_{n-2} c_2 - \Delta t^3 \alpha_{n-3} d_2 \dots]$$

$$\vdots$$

$$\lambda^{p-n} [(-1)^n - \Delta t \alpha_{n-1} b_{p-n} - \Delta t^2 \alpha_{n-2} c_{p-n} - \Delta t^3 \alpha_{n-3} d_{p-n} \dots]$$

$$+ \lambda^{p-n-1} [-\Delta t \alpha_{n-1} b_{p-n-1} - \Delta t^2 \alpha_{n-2} c_{p-n-1} \dots \dots \dots]$$

$$[-\Delta t \alpha_{n-1} b_p - \Delta t^2 \alpha_{n-2} c_p - \Delta t^3 \alpha_{n-3} d_p \dots \dots \dots] \quad (12)$$

where the control $q(t, k)$ is treated as a non-homogeneous term. The characteristic polynomial is a complex coefficient polynomial in the dual variable k . Its roots determine stability of the finite difference scheme.

III. STABILITY OF THE DIFFERENCE SCHEME

The characteristic polynomial (12) is derived for constant coefficients in (1). The theory of stability for difference methods is developed for constant coefficient, and often 'pure' initial-value problems [2]. Hence it is with restrictions that this theory be used to establish stability for the difference approximation to (1), although it is justified when bounds can be used for the coefficients in (1). For $\Delta x = g(\Delta t)$, where g is a continuous function with $g(0) = 0$, the stability criterion used is: Von Neumann's Theorem [2]. A necessary condition for stability of a difference scheme is that the roots of the characteristic equation must be:

$$|\lambda_i| \leq 1 + O(\Delta t) \quad \text{for} \quad \begin{cases} 0 < \Delta t \leq \Delta t_0 \\ \text{All values } i \\ i=1,2,3,\dots,p \end{cases} \quad (13)$$

Therefore if the homogeneous part of the characteristic polynomial given by (12) has its roots inside the unit circle it represents a stable formulation. If the a_i result from definite difference schemes and have the desired properties subsequently discussed, then for a fixed Δx , Δt can be made small enough so that the non-homogeneous terms satisfy (13). The known technique for testing the roots of (12) is the Schur-Cohn criterion, which is valid for complex coefficient polynomials.

Schur-Cohn Criterion [3] If for the polynomial

$$p(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 \dots a_{p-1}\lambda^{p-1} + a_p\lambda^p \quad (14)$$

all the determinants: (for $s=1,2,3,\dots,p$)

$$\Delta_s = \begin{vmatrix} a_0 & 0 & 0 & 0 & a_p & a_{p-1} & a_{p-2} & \dots & a_{p-s+1} \\ a_1 & a_0 & 0 & 0 & 0 & a_p & a_{p-1} & & . \\ a_2 & a_1 & a_0 & & 0 & 0 & a_p & & . \\ . & & & a_0 & 0 & . & & a_p & a_{p-1} \\ a_{s-1} & a_{s-2} & & a_1 & a_0 & . & & 0 & a_p \\ \hline \bar{a}_p & 0 & 0 & 0 & \bar{a}_0 & \bar{a}_1 & \bar{a}_2 & \dots & \bar{a}_{s-1} \\ \bar{a}_{p-1} & \bar{a}_p & 0 & 0 & 0 & \bar{a}_0 & \bar{a}_1 & & \bar{a}_{s-2} \\ \bar{a}_{p-2} & \bar{a}_{p-1} & \bar{a}_p & & 0 & 0 & \bar{a}_0 & & . \\ . & & & \bar{a}_p & 0 & & & \bar{a}_0 & \bar{a}_1 \\ \bar{a}_{p-s+1} & \bar{a}_{p-s+2} & \dots & \bar{a}_{p-1} & \bar{a}_p & & & 0 & \bar{a}_0 \end{vmatrix} \quad (15)$$

are different from zero, then $p(\lambda)$ has no zeros on the unit circle $|\lambda| = 1$ and r zeros inside this circle; r being the number of variations in sign of the determinant sequence: $1, \Delta_1, \Delta_2, \Delta_3, \dots, \Delta_p$.

To have all roots inside the unit circle:

$$\Delta_s > 0 \quad \text{for } s=1,3,5,\dots \quad \text{odd determinants} \quad (16)$$

$$\Delta_s > 0 \quad \text{for } s=2,4,6,\dots \quad \text{even determinants}$$

Equation (16) is the condition which the difference approximation to (1) must satisfy, while it is seen that if $\Delta s > 0$ for all s , then every root lies outside the unit circle. If all roots are outside the unit circle, then $|\frac{1}{\lambda}| < 1$ and the difference equation is stable when solved backwards in time. Hence the difference equation to the adjoint equation, is formulated to have all roots outside the unit circle since it is a final value problem and its solution is found by 'marching backwards' from the known n final rows.

IV. SOLVING THE OPTIMAL CONTROL PROBLEM

Consider that definite difference schemes are used for the space difference approximations to (1) and its adjoint equation are chosen to have the eigenvalues of homogeneous parts inside and outside the unit circle respectively. Then an iterative scheme is used to solve for the mesh values. The iterative scheme employed here is an extension to several dimensions of a technique utilized in [4], and is performed on (1) and its adjoint q (say that e is eliminated).

Method of Solution

- Use the boundary and initial conditions given for u to solve its difference equation in the forward time direction to obtain $u(1)$ on the mesh with all $q(0) = 0$
- Use the final values, boundary conditions and the values of $u(1)$, to sweep backwards in time solving for $q(1)$ for the entire mesh
- Continue the process of solving for $u(r)$ using the latest estimate $q(r-1)$ by sweeping forward and solve for $q(r)$ by sweeping backwards until some suitable convergence criterion has been achieved.

The convergence criterion used for this process is the value of the performance index, since it is a quadratic function expressed for the entire mesh. When the value of J obtained for several successive iterations does not vary more than one percent, for instance, then the values for the dependent variable and control are considered a suitable result.

V. EXAMPLE 1

Consider the system described the PDE:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + (1 + \frac{x^2}{2}) \frac{\partial^2 u}{\partial x \partial t} + q(t, x) \quad (1a)$$

which has the given boundary conditions:

$$u(0, x) = \frac{\partial u}{\partial t}(0, x) = u(t, 0) = u(t, 1) = 0 \quad (1b)$$

It is intended to force the system toward a desired state:

$$u_D = x(1-x)\epsilon^t \quad (1c)$$

by means of the control $q(t, x)$ to minimize the performance index:

$$J = \int_{t=0}^{t=.2} \int_{x=0}^{x=1} \{ [u - u_D]^2 + \frac{q^2}{W} \} \partial x \partial t \quad (1d)$$

Carrying out the steps outlined in section I results in a partial differential equation for q , boundary and final conditions:

$$\frac{\partial^2 q}{\partial t^2} = \frac{\partial^2 q}{\partial x^2} + \left(1 + \frac{x}{2}\right) \frac{\partial^2 q}{\partial x \partial t} + x \frac{\partial q}{\partial t} + W(u_D - u) \quad (1e)$$

$$q(t = .2, x) = \frac{\partial q}{\partial t}(t = .2, x) = 0 \quad q(t, 0) = q(t, 1) = 0 \quad (1f)$$

It is first seen that the homogeneous part of (1a) is:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \left(1 + \frac{x}{2}\right) \frac{\partial^2 u}{\partial x \partial t} = \alpha'_0 u + \alpha'_1 \frac{\partial u}{\partial t} \quad (1g)$$

Let the space derivatives in (1g) be approximated by difference schemes, the space dependency Fourier transformed, and a general three time-step approximation to be used to obtain the 'point-wise' valid equation:

$$[1 + b_1 \Delta t \alpha_1 + c_1 \Delta t^2 \alpha_0] u[(m+2)\Delta t, k] + [-2 + b_2 \Delta t \alpha_1 + c_2 \Delta t^2 \alpha_0] u[(m+1)\Delta t, k] + [1 + b_3 \Delta t \alpha_1 + c_3 \Delta t^2 \alpha_0] u[m\Delta t, k] = 0 \quad (1h)$$

In order to be consistent to (1g) the b_i and c_i must satisfy:

$$b_1 + b_2 + b_3 = 0 \quad 2b_1 + b_2 = -1 \quad c_1 + c_2 + c_3 = -1 \quad (1i)$$

The characteristic polynomial after eliminating b_2 , b_3 , and c_3 is:

$$p_H(\lambda) = \lambda^2 [1 + b_1 \Delta t \alpha_1 + c_1 \Delta t^2 \alpha_0] + \lambda [-2 - \Delta t (1 + 2b_1) \alpha_1 + c_2 \Delta t^2 \alpha_0] + [1 + \Delta t (1 + b_1) \alpha_1 - \Delta t^2 (1 + c_1 + c_2) \alpha_0] = a_2 \lambda^2 + a_1 \lambda + a_0 \quad (1j)$$

Note the quadrants of the Schur-Cohn criterion commute, so the $p-1$ lower order determinants are the principal minors of the $p \times p$ determinant. For equation (1j) these determinants are found from:

$$\Delta_2 = \begin{vmatrix} a_0 & \bar{a}_0 - a_2 & \bar{a}_2 & | & a_0 & \bar{a}_1 - a_1 & \bar{a}_2 \\ \hline & & \Delta_1 & | & & & \\ \hline \bar{a}_0 & \bar{a}_1 - a_1 & a_2 & | & a_0 & \bar{a}_0 - a_2 & a_2 \end{vmatrix} \quad (1k)$$

The coefficients of (1j) are introduced into (1k) so that the expansion of Δ_1 is:

$$\Delta_1 = \Delta t (\alpha_1 + \bar{\alpha}_1) - \Delta t^2 (1 + 2c_1 + c_2) (\alpha_0 + \bar{\alpha}_0) + \Delta t^2 (1 + 2b_1) \alpha_1 \bar{\alpha}_1 + \Delta t^3 \{ (1 + b_1) (-1 - c_1 - c_2) - b_1 c_1 \} \{ \alpha_0 \bar{\alpha}_1 + \bar{\alpha}_0 \alpha_1 \} + \Delta t^4 (1 + 2c_1 + 2c_2 + c_1^2 + 2c_1 c_2) \alpha_0 \bar{\alpha}_0 \quad (1l)$$

Since Δ_1 must be negative, to have the correct sign as $\Delta t \rightarrow 0$:

$$\Delta_1 \approx \Delta t (\alpha_1 + \bar{\alpha}_1) = 2\Delta t \operatorname{Re} \{ (1 + \frac{x}{2}) \frac{\partial}{\partial x} \} \leq 0 \quad (1m)$$

Therefore the approximation to $\frac{\partial}{\partial x}$ must have a Fourier transform with a negative real part as given by (9a), which is a forward difference scheme.

If $\Delta_2 > 0$ as $\Delta t \rightarrow 0$, then the lowest power Δt term must be the correct sign as $\Delta t \rightarrow 0$. This term in the expansion of (1k) is:

$$\Delta_2 \approx \Delta t^4 (\alpha_0 - \bar{\alpha}_0)^2 - \Delta t^4 (\alpha_1 + \bar{\alpha}_1) (\alpha_0 \bar{\alpha}_1 + \bar{\alpha}_0 \alpha_1) \geq 0 \quad (1n)$$

Since α_0 represents the Fourier transform of the difference scheme to $\frac{\partial^2}{\partial x^2}$, (8) is a definite difference scheme which yields $\operatorname{Re} \{ \} \leq 0$ and $\operatorname{Im} \{ \} = 0$. Hence:

$$\Delta_2 \approx -4\Delta t^4 \operatorname{Re} \{ \alpha_0 \} [\operatorname{Re} \{ \alpha_1 \}]^2 \geq 0 \quad (1p)$$

There is no need to specify non-zero values to b_1, c_1 and c_2 since the lowest power Δt term of Δ_1 and Δ_2 are non-zero. In some circumstances use of these coefficients permits larger time steps.

The difference approximation to (1a) fixed by the choice of difference schemes is then solved explicitly for $u[(m+2)\Delta t, n\Delta x]$:

$$\begin{aligned} & \frac{u[(m+2)\Delta t, n\Delta x] - 2u[(m+1)\Delta t, n\Delta x] + u[m\Delta t, n\Delta x]}{\Delta t^2} \\ &= \frac{u[m\Delta t, (n+1)\Delta x] - 2u[m\Delta t, n\Delta x] + u[m\Delta t, (n-1)\Delta x]}{\Delta x^2} \\ &+ \left\{1 + \frac{n^2 \Delta x^2}{2}\right\} \left\{ \frac{u[(m+1)\Delta t, (n+1)\Delta x] - u[(m+1)\Delta t, n\Delta x]}{\Delta x \Delta t} \right. \\ &\quad \left. - \frac{u[m\Delta t, (n+1)\Delta x] + u[m\Delta t, n\Delta x]}{\Delta x \Delta t} \right\} + q[m\Delta t, n\Delta x] \quad (1q) \end{aligned}$$

Next, to consider the difference approximation to (1e), if the time polynomial is retained at three time steps, this again results in a second order characteristic equation. The homogeneous part of (1e) has the same form as (1g), therefore equations (1j), (1k), (1l), and (1n) also describe the three time step approximations to (1e).

In (1l) and (1n) it is seen that the lowest power Δt term is independent of the unspecified b_1 and c_1 . Hence if:

$$\Delta_1^{\sim} = \Delta t^4 (\alpha_0 - \bar{\alpha}_0)^2 - \Delta t^4 (\alpha_1 + \bar{\alpha}_1) \geq 0 \quad (1r)$$

$$\Delta_2^{\sim} = \Delta t^4 (\alpha_0 - \bar{\alpha}_0)^2 - \Delta t^4 (\alpha_1 + \bar{\alpha}_1) (\alpha_0 \bar{\alpha}_1 + \bar{\alpha}_0 \alpha_1) \geq 0 \quad (1s)$$

are observed, then an explicit formulation solved for the lowest time step will be stable as $\Delta t \rightarrow 0$. To accomplish this, the space derivative part

of: $(1 + \frac{x}{2}) \frac{\partial^2 q}{\partial x \partial t}$ must be approximated by a backwards difference scheme for (1r) to be satisfied. This scheme is given by (9b).

To satisfy equation (1s), note that α_0 is a second derivative for which the definite difference scheme has $\text{Re}\{\} \leq 0$, so that equation (1p) holds and the second root of the difference scheme to (1e) will be outside the unit circle. The resulting difference approximation to (1e) is:

$$\begin{aligned} & \frac{q[(m+2)\Delta t, n\Delta x] - 2q[(m+1)\Delta t, n\Delta x] + q[m\Delta t, n\Delta x]}{\Delta t^2} \\ &= \frac{q[(m+2)\Delta t, (n+1)\Delta x] - 2q[(m+2)\Delta t, n\Delta x] + q[(m+2)\Delta t, (n-1)\Delta x]}{\Delta x^2} \\ &+ \left\{1 + \frac{n^2 \Delta x^2}{2}\right\} \left\{ \frac{q[(m+2)\Delta t, n\Delta x] - q[(m+2)\Delta t, (n-1)\Delta x]}{\Delta x \Delta t} \right. \\ &\quad \left. - \frac{q[(m+1)\Delta t, n\Delta x] + q[(m+1)\Delta t, (n-1)\Delta x]}{\Delta x \Delta t} \right\} + n\Delta x \frac{q[(m+2)\Delta t, n\Delta x] - q[(m+1)\Delta t, n\Delta x]}{\Delta t} \\ &\quad + W\{u_D[m\Delta t, n\Delta x] - u[m\Delta t, n\Delta x]\} \quad (1t) \end{aligned}$$

In this expression it can be seen that a formulation for $q[m\Delta t, n\Delta x]$ is explicit in terms of quantities known at the following time rows, i.e. the $(m+1)\Delta t$ and $(m+2)\Delta t$ rows in time.

The difference schemes have now been formulated, and for a given Δx size it remains to choose the appropriate Δt step size. This is easily done for the homogeneous part of the difference schemes, but the quantity $O(\Delta t)$ of (13) is difficult to estimate.

A starting value from which the time step may be reduced is given by (1m):

$$\Delta_1^{\sim} = 2\Delta t \left| \text{Re} \left\{ \left(1 + \frac{x}{2}\right) \frac{\partial}{\partial x} \right\} \right|_{\max} < 1$$

The maximum of this expression occurs at $k\Delta x = \pi$ and at $x=1$, and reduces to:

$$3 \frac{\Delta t}{\Delta x} < 1$$

This criterion is suitable since definite difference schemes are symmetric about $k\Delta x = \pi$ and the lowest power Δt term tends to dominate the Schur-Cohn determinants. The non-homogeneous part of (1a) is the control q , while the non-homogeneous part of (1e) is:

$$x \frac{\partial q}{\partial t} + w(u_D - u)$$

Such that a trial and error process may be necessary to assure that these terms are sufficiently small. However since the homogeneous part is stable, the Δt step size can always be reduced to a value which results in a stable calculation.

Numerical results are presented in figures 1-3. In figure 2 it may be seen that for $W = 50$ the non-homogeneous part of (1e) is much larger and hence the convergence of the iterative process is slower. Note in figure 3 that the desired state is more closely attained because the relative cost of control energy is less.

IX. CONCLUSIONS

The iterative method of solution was demonstrated to converge if a stable difference formulation is used for both the dependent variable and the adjoint equation. The uniqueness of the solution is inferred from the fact that the difference formulations are consistent, stable, and the solution satisfies the partial differential equations as well as the boundary conditions.

The finite difference formulations for the optimal control problem discussed here also apply to the simpler case of a difference equation for an initial-value type partial differential equation. Many of the principles presented herein extend to the multiple dimension case where x is a vector.

References

- (1) Forsythe, A. R., "Calculus of Variations". Dover Publications, Inc., New York, 1960.
- (2) Richtmyer, R. D., "Difference Methods for Initial-Value Problems", Interscience Publishers, New York, 1957.
- (3) Marden, M., "The Geometry of the Zeros". American Mathematical Society, New York, 1949.
- (4) Levine, M. D., "Trajectory Optimization Using the Newton-Raphson Method", Automatica, Volume 3, Numbers 3/4, January 1966.

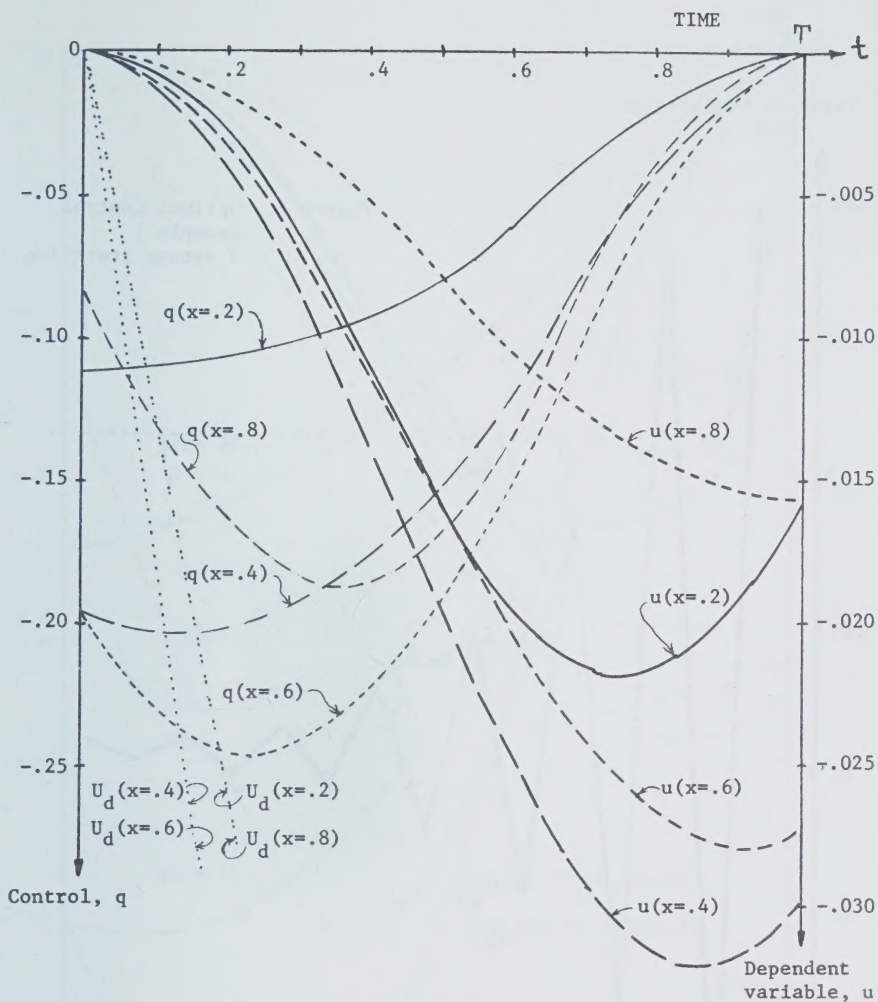
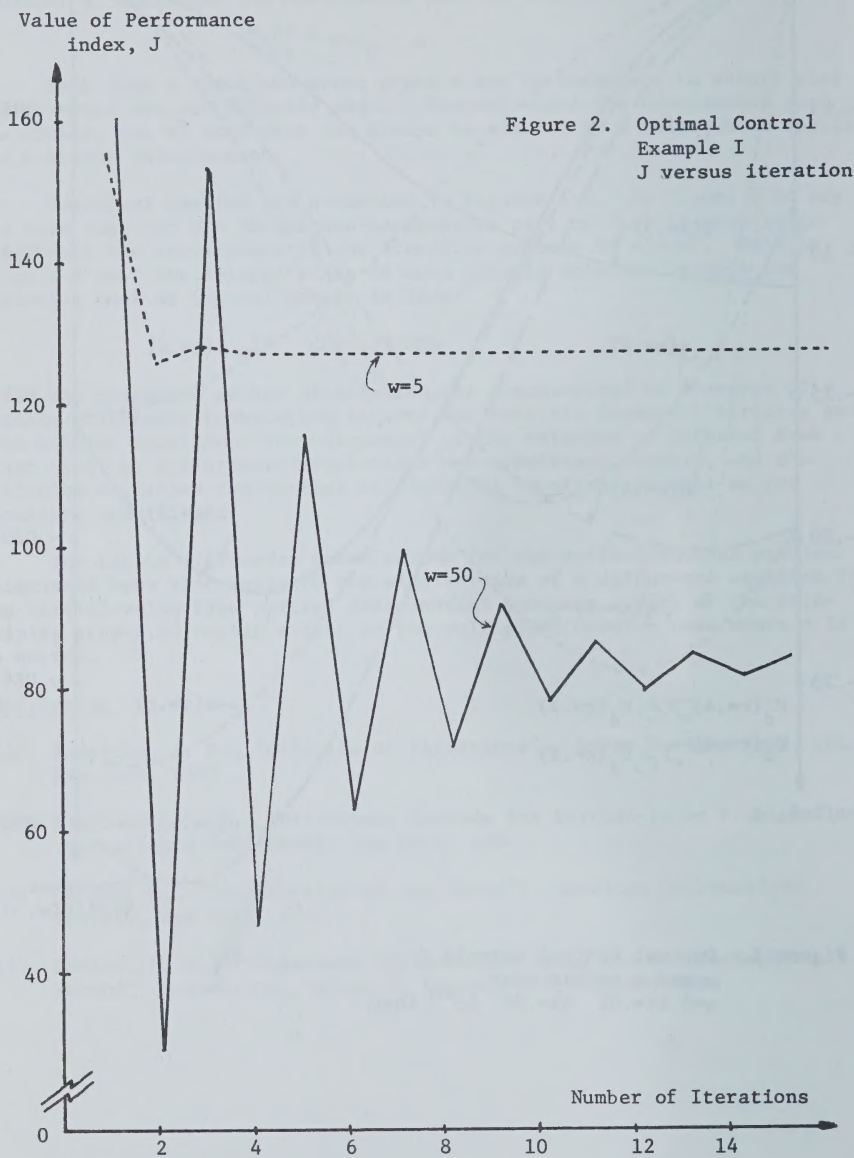


Figure 1. Optimal Control Example I
 u and q versus time
 $w=5$ $\Delta t=.01$ $\Delta x=.04$ 15th iter.



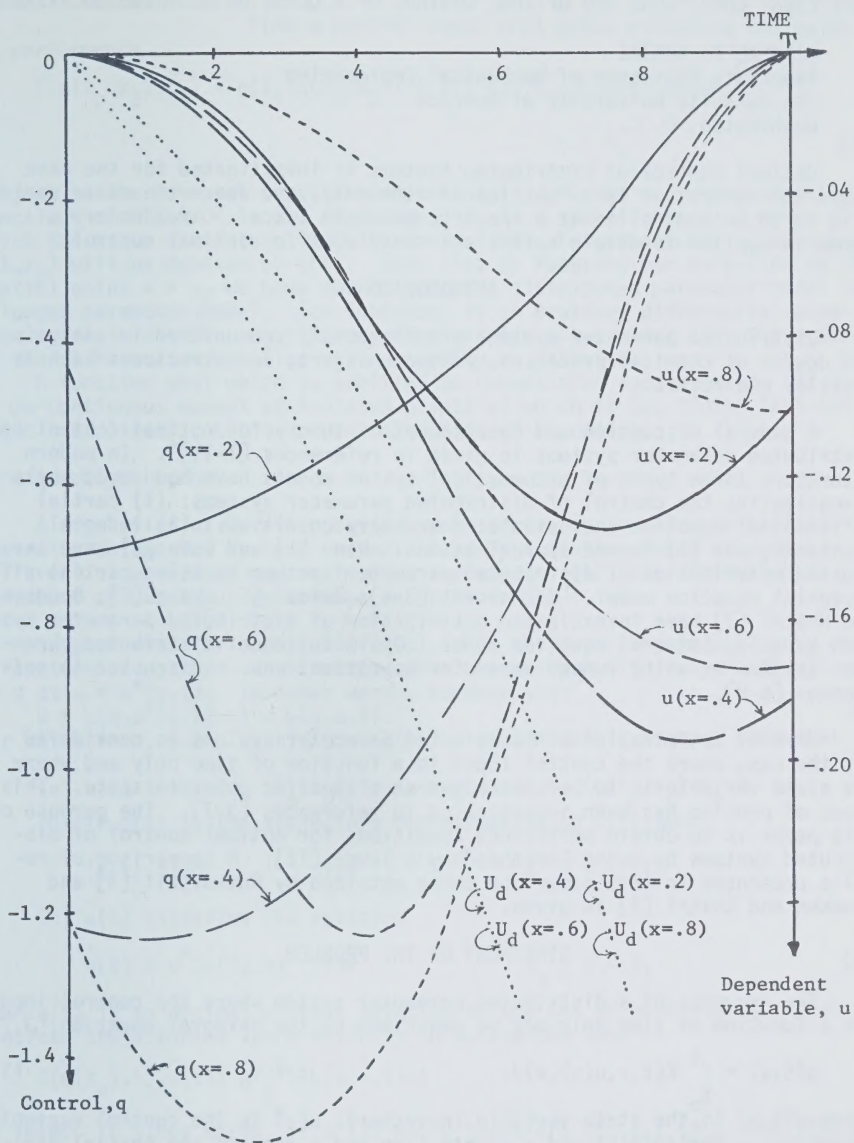


Figure 3. Optimal Control Example I
 u and q versus time
 $w=50 \Delta t = .01 \Delta x=.04$ 15th iter.

HIRALAL C. KHATRI

Associate Professor of Mechanical Engineering
The Catholic University of America
Washington, D. C.

Optimal Control of Distributed Systems is investigated for the case where the controller is a function of time only, and where the state variable is to be controlled at a specific point in space. Caratheodory's lemma is applied to obtain sufficient conditions for optimal control.

INTRODUCTION

Distributed parameter systems are frequently encountered in analysis and design of chemical processes, heat exchangers, and structures such as missiles and rockets.

A general discussion and development of theory for optimal control of distributed parameter systems is given in references [1-11]*. In modern literature three types of mathematical system models have been used while investigating the control of distributed parameter systems: (1) partial differential equation and associated boundary conditions; (2) integral equations; and (3) lumped approximations. Wang [1] and Lure [2] have investigated optimization of distributed parameter systems by using partial differential equation model. Butkovskii [3-4], Sakawa [5], Wang [6], Goodson and Khatri [7] have investigated optimization of distributed parameter systems by using integral equation model. Optimization of distributed parameter systems by using lumped parameter approximations is discussed in references [6-8].

Herein, optimization of distributed parameter systems is considered for the case where the control input is a function of time only and where the state variable is to be controlled at a specific point in space. This class of problem has been investigated in references [3-7]. The purpose of this paper is to obtain sufficient conditions for optimal control of distributed systems by using Caratheodory's lemma [12]. A comparison of results presented in this paper and those obtained by Butkovskii [3] and Goodson and Khatri [7] is given.

STATEMENT OF THE PROBLEM

The dynamics of a distributed parameter system where the control inputs are a function of time only may be described by the integral equation [3,7]

$$q(t, x) = \int_{t_0}^t K(t, \tau, u(\tau), x) d\tau \quad t > t_0 \quad (1)$$

where $q(t, x)$ is the state variable (n -vector), $u(\tau)$ is the control variable (m -vector). Variables t and τ denote time and x denotes the spatial variable. The integrand $K(t, \tau, u(\tau), x)$ is an n -vector such that

$$K(t, \tau, u(\tau), x) = 0 \quad \text{for } \tau > t \quad (2)$$

where 0 is a null vector. It is assumed that $K(t, \tau, u(\tau), x)$ and its derivatives with respect to t and $u(\tau)$ are continuous for $t > \tau$.

*Numbers in square parentheses designate references at the end of this paper.

Optimal Control Problem.

Find a control input $u(t)$ which minimizes the index of performance

$$I(q(t_0, x_2), t_1) = g(t_1, q(t_1, x_2)) + \int_{t_0}^{t_1} L(q(t, x_2), u(t), t) dt$$

$$t_1 > t_0 \quad (3)$$

subject to the integral equation constraint (1). L and g are non-negative functions of Class C^2 in all arguments. x_2 is a specific spatial point at which the state variable q is to be controlled. For simplicity of notation, $q(t, x_2)$ will be denoted by $q(t)$. Note that by focusing our attention on the spatial point $x = x_2$ we have not reduced the distributed parameter model to a lumped parameter model. For instance, if an ordinary differential equation model is used to describe relation between $q(t, x_2)$ and $u(t)$ the order of the differential equation will generally be infinite.

A function $u(t)$ which is admitted to competition in minimizing (3) is to be continuous except at isolated points at which it has finite left- and right-hand limits. Let Ω denote this class of functions.

Caratheodory's Lemma

A discussion of Caratheodory's lemma and its application to optimal control problems for systems characterized by a finite number of ordinary differential equations may be found in references [12,13]. Following is a statement of Caratheodory's lemma as given in [12]:

Suppose that for each point (q, t) in Q , the function $L(q, \omega, t)$ has, as a function of ω , zero as its unique absolute minimum with respect to all ω in Ω at $\omega = u^*(q, t)$. In other words, suppose that

$$0 = L(q, u^*(q, t), t) < L(q, \omega, t) \quad (4)$$

for all ω in Ω , $\omega \neq u^*(q, t)$. Let $u(t)$ be an admissible control such that:

(a) $\hat{u}(t)$ transfers (q_0, t_0) to a target set S ; $(q(t_0, x_2) = q_0)$.

(b) If $\hat{q}(t)$ is the trajectory corresponding to $\hat{u}(t)$ then

$$(\hat{q}(t), t) \in Q \quad \text{for} \quad t_0 \leq t \leq t_1 \quad (5)$$

(c) $\hat{u}(t)$ satisfies the relation

$$\hat{u}(t) = u^*(\hat{q}(t), t) \quad \text{for} \quad t_0 \leq t < t_1 \quad (6)$$

Then $\hat{u}(t)$ is an optimal control relative to the set of controls $u(t)$ that generate trajectories lying entirely in Q , and the cost

$$J(\hat{q}(t_0), t_0, \hat{u}(t)) = \int_{t_0}^{t_1} L(\hat{q}, \hat{u}, t) dt$$

is zero. In other words, if $u'(t)$ which transfers (q_0, t_0) to S such that $(q'(t), t) \in Q$ where $q'(t)$ is the trajectory corresponding to u' , then

$$0 = J(q_0, t_0, \hat{u}(t)) \leq J(q_0, t_0, u'(t)) \quad (7)$$

The proof of this lemma is not given here since it is readily available in [12,13].

In order to apply the Caratheodory's lemma to our optimal control problem it is necessary to modify our index of performance such that its integrand satisfies the condition given in equation (4). The following

modification of the index of performance is very similar to that given by Kalman [13] for systems described by a finite number of ordinary differential equations.

Define $V(q(t), t)$ to be a scalar function (finite valued) that is of class C^2 in $q(t)$ and t and subject to the condition

$$V(q(t_1), t_1) = g(q(t_1), t_1) \quad (8)$$

Then

$$\int_{t_0}^{t_1} [V_t + \langle V_q, \dot{q} \rangle] dt = V(q(t_1), t_1) - V(q(t_0), t_0) = g(q(t_1), t_1) - V(q(t_0), t_0) \quad (9)$$

along any motion of equation (1) connecting the phase $(q(t_0), t_0)$ with the phase $(q(t_1), t_1)$ on S . Here $\langle \cdot, \cdot \rangle$ denotes inner product. $V_t = \frac{\partial V}{\partial t}$ and V_q is an n -vector

$$V_q = \left[\frac{\partial V}{\partial q_1}, \dots, \frac{\partial V}{\partial q_n} \right] \quad (10)$$

where q_i are the elements of vector q . $\dot{q}$ denotes the vector $\left[\frac{dq_1}{dt}, \dots, \frac{dq_n}{dt} \right]$

For our problem $q(t_0)$ is zero.

Substituting for $\dot{q}$ from equation (1) and (9) and simplifying gives

$$g(q(t_1), t_1) - V(q(t_0), t_0) = \int_{t_0}^{t_1} V_t dt + \int_{t_0}^{t_1} dt \int_{t_0}^{t_1} d\tau \langle V_q(q(t), t), K_t(t, \tau, u(\tau)) \rangle \quad (11)$$

where K_t denotes $\left[\frac{\partial^2 K}{\partial t^2}, \dots, \frac{\partial^2 K}{\partial t^n} \right] \dots$

For reasons that will appear later on, the order of integration with respect to τ and t in equation (11) is interchanged, thus [7]

$$\begin{aligned} g(q(t_1), t_1) - V(q(t_0), t_0) &= \int_{t_0}^{t_1} V_t dt + \int_{t_0}^{t_1} dt \int_{t_0}^{t_1} d\tau \langle V_q(q(t), t), K_t(t, \tau, u(\tau)) \rangle \\ &= \int_{t_0}^{t_1} V_t dt + \int_{t_0}^{t_1} dt \int_t^{t_1} d\tau \langle V_q(q(\tau), \tau), K_\tau(\tau, t, u(t)) \rangle \end{aligned} \quad (12)$$

where the variables τ and t have been interchanged in the last term. This interchange of order of integration is valid provided the value of the integral is finite [14]. (Note V has been assumed finite)

In order to apply the Caratheodory's lemma to the optimal control problem defined by equations (1) and (3) consider the integral

$$\begin{aligned} J &= \int_{t_0}^{t_1} [L(q(t), u(t), t) + V_t + \int_t^{t_1} \langle V_q(\tau), K_\tau(\tau, t, u(t)) \rangle d\tau] dt \\ &= I(q(t_0), t_1) - V(q(t_0), t_0) \end{aligned} \quad (13)$$

Since $V(q(t_0), t_0)$ is independent of the controller u , minimization of J is equivalent to minimization of the original index of performance I in equation (3).

Following the procedure for optimal control of lumped parameter systems let a costate vector p be defined by

$$p = V_q \quad (14)$$

and a scalar H by

$$H(q, p, t, u) = L(q, u, t) + \int_t^{t_1} \langle p(\tau), K_\tau(\tau, t, u(\tau)) \rangle d\tau \quad (15)$$

Assume that H has a unique absolute minimum for each t with respect to $u(t) \in \Omega$ at the point

$$u^*(t) = c(q, p, t) \quad (16)$$

where c is an m -vector. Normally, c will be a functional of p and a function of q and t . The scalar function H^* defined by the following equation is considered as the Hamiltonian of the problem.

$$\begin{aligned} H^*(q, p, t) &= \min_{u(t) \in \Omega} H(q, p, t, u) \\ &= L(q, c(q, p, t), t) + \int_t^{t_1} \langle p(\tau), K_\tau(\tau, t, c(q, p, t)) \rangle d\tau \end{aligned} \quad (17)$$

The arbitrary function V is now assumed to satisfy the integro-differential equation

$$V_t + H^*(q, V_q, t) = 0 \quad (18)$$

with the boundary condition (8). Equation (18) is the Hamilton-Jacobi equation of our problem relative to the phase space Q .

In view of equation (15), equation (13) takes the form:

$$J(q(t_0), t_0) = \int_{t_0}^{t_1} [H(q, V_q, t) + V_t] dt \quad (19)$$

Since V is assumed to satisfy equation (18), the integrand of equation (19) is identically equal to zero when H equals its absolute minimum H^* . Thus, an index of performance J has been constructed whose integrand satisfies one of the hypotheses of Caratheodory's lemma, equation (4). Hence the controller which minimizes J , and therefore I , is given by

$$u^*(t) = c(q, V_q, t) \quad (20)$$

where V satisfies equation (18) and c minimizes H , equation (17) provided $u^* \in \Omega$ and $q^* \in Q$.

When the control variable u is the optimal u^* , the value of J in equation (19) is zero and thus from equation (13)

$$V(q^*(t_0), t_0) = I(q^*(t_0), t_0) \quad (21)$$

where $I(q^*(t_0), t_0)$ is the value of the original index of performance (3) and q^* is the trajectory corresponding to the optimal control u^* .

Note that H in equation (15) is a function of q , t and u and a functional of p . If the order of integration in equation (12) had not been interchanged, it would have been necessary to define H as a functional of u and a function of p . This would have required minimization of a functional of u instead of a function of u in order to obtain H^* , equation (17). In most instances, minimization of a function is easier than minimization of a functional and that is why the order of integration was interchanged.

The results are summarized by the theorem:

Suppose that H defined by equation (15) has an absolute minimum relative to Q given by equation (16). Let $u^*(t)$ be an admissible control such that:

- (a) $u^*(t)$ transfers (q_0, t_0) to S
- (b) If $q^*(t)$ is the trajectory corresponding to $u^*(t)$, then $(q^*(t), t) \in Q$ for $t_0 \leq t \leq t_1$
- (c) There is a solution $V(q(t), t)$ of the Hamilton-Jacobi equation (18) satisfying the boundary condition (8) such that $u^* = c(q^*, V(q^*, t), t)$ for $t_0 \leq t \leq t_1$.

Then u^* is an optimal control relative to the set of control $u \in \Omega$ that generates trajectories lying entirely in Q and

$$I(q^*(t), t) = V(q^*(t), t) \quad \text{for} \quad t_0 \leq t \leq t_1. \quad (22)$$

Note that all statements are made relative to the region Q and thus the above theorem is a local sufficiency condition for optimality. Reference [12] has an excellent discussion on limitations and applicability of this theorem to the problem of optimal control of lumped systems and the same discussion applies for the distributed system under consideration.

Canonical Equations

A set of relations analogous to the canonical equations of the classical variational calculus are now presented.

Substituting for I from equation (3) in equation (22) gives

$$V(q^*(t), t) = g(t_1, q^*(t_1)) + \int_{t_0}^{t_1} L(q^*(\tau), u^*(\tau), \tau) d\tau \quad (23)$$

Differentiation of equation (14) with respect to t and a subsequent application of the chain rule [15] yield

$$\frac{dp}{dt} = \frac{\partial}{\partial q} \left[\frac{dV}{dt} \right] \quad (24)$$

where

$$\frac{dp}{dt} = \left[\frac{dp_1}{dt}, \dots, \frac{dp_n}{dt} \right] \quad (25)$$

and

$$\frac{\partial}{\partial q} \left[\frac{dV}{dt} \right] = \left[\frac{\partial}{\partial q} \frac{dV}{dt}, \dots, \frac{\partial}{\partial q_n} \frac{dV}{dt} \right] \quad (26)$$

Evaluating $\frac{dV}{dt}$ from equation (23) and substituting the results in equation (24) one obtains

$$\frac{dp}{dt} = -L_q(q^*(t), u^*(t), t) \quad (27)$$

where

$$L_q = \left[\frac{\partial L}{\partial q_1}, \dots, \frac{\partial L}{\partial q_n} \right]. \quad (28)$$

Differentiation of H^* , equation (17), with respect to q yields (note H^* is a function of q)

$$H_q^* = L_q + H_c^* c_q \quad (29)$$

where

$$H_q^* = \left[\frac{\partial H^*}{\partial q_1}, \dots, \frac{\partial H^*}{\partial q_n} \right] \quad (30)$$

$$H_c^* = \left[\frac{\partial H^*}{\partial c_1}, \dots, \frac{\partial H^*}{\partial c_m} \right] \quad (31)$$

and

$$c_q = \begin{bmatrix} \frac{\partial c_1}{\partial q_1} & \dots & \frac{\partial c_1}{\partial q_n} \\ \vdots & & \vdots \\ \frac{\partial c_m}{\partial q_1} & \dots & \frac{\partial c_m}{\partial q_n} \end{bmatrix} \quad (32)$$

If $c(q, p, t)$ is interior to Ω then H_C^* equals zero since c minimizes H . If c lies on the boundary of Ω then H_C^* is zero provided the boundary of Ω is smooth [13]. Assuming that the boundary of Ω is smooth, substituting for L_q from (29) in equation (27) yields

$$\frac{dp}{dt} = -H_q^* \quad (33)$$

This, of course, is a familiar equation from the classical variational calculus.

In order to obtain a relation between H^* and q it is necessary to differentiate H^* with respect to p . But H^* is a functional of p . The functional derivative of H^* with respect to p is given by the operator [16]

$$\frac{\delta H^*}{\delta p} = K_\tau(\tau, t, c(q(t), p(t), t)) \quad (34)$$

where $\frac{\delta H^*}{\delta p}$ denotes functional derivative. The fact that H_C^* equals zero has been used. Reference [16] has a critical study of functional derivatives and their applications to variational calculus.

Integration of equation (34) with respect to t yields

$$\int_0^\tau \frac{\delta H^*}{\delta p} dt = \int_0^\tau K_\tau(\tau, t, c(q(t), p(t), t)) dt = \dot{q}(\tau) \quad (35)$$

This result is analogous to one of the canonical equations of the classical variational calculus.

Canonical equations (33) and (35) can be derived from the Hamilton-Jacobi equation, (18), however the algebra gets complicated because (18) is an integro-differential (partial) equation. It may be pointed out that if the order of integration in equation (15) had not been interchanged, H would have been a function of p and instead of equation (35) we would have obtained

$$H_p^* = \dot{q} \quad (36)$$

In this case equation (33) would remain the same. However, as pointed out earlier, H would then be a functional of u which complicates the evaluation of H^* .

Results presented here are now compared with those given by Butkovskii in [3] for the class of problems under consideration.

The essential results of Butkovskii given in reference [3] may be paraphrased as follows. Consider a system described by equations (1) and (2) and the index of performance given by (3) with g equal to zero. Let $u = u(t) \in \Omega$ ($t_0 \leq t \leq t_1$), for which $q(t_1) \in S$. For the optimum control $u^*(t)$ and the corresponding trajectory $q^*(t)$ there should exist a constant nonzero vector $\lambda = (\lambda_1, \dots, \lambda_n)$ together with a constant $\lambda_0 \leq 0$ such that, for almost all t , $t_0 \leq t \leq t_1$, the function

$$H = \lambda_0 L(t, q(t), u(t)) + \lambda_0 \int_t^{t_1} \langle L_q(q(\tau), u(\tau), \tau), K(\tau, t, u(t)) \rangle d\tau + \langle \lambda, K(t_1, t, u(t)) \rangle \quad (37)$$

of the variable $u(t)$, $u(t) \in \Omega$, attains its maximum at the point $u^*(t)$. Moreover, the transversality condition at the point $q(t_1)$ must be satisfied.

An inspection of equations (15) and (37) shows that in order to compare the two H functions it is necessary to carry out integration by parts in equation (15), thus

$$H(q, p, t, u) = L(q, u, t) - \int_t^{t_1} \langle \dot{p}(\tau), K(\tau, t, u(t)) \rangle d\tau + \langle p(t_1), K(t_1, t, u(t)) \rangle \quad (38)$$

where the condition on K given in equation (2) has also been used.

In view of equation (27) it is seen that minimization of H given in equation (38) is equivalent to maximization of H given in equation (37). However, Butkovskii's results are valid without the strong smoothness assumption on V and are therefore applicable to a wider class of problems. Example

Optimal control of a heating system that was given in reference [7] is now considered to illustrate the procedure for calculating the optimal controller.

Briefly, the mathematical statement of the problem given in [7] is as follows. Determine the control variable $u(t)$ which minimizes the following index of performance:

$$I = \int_0^\infty \{ [u(t) - u_s]^2 + [q_d - q(t)]^2 \} dt \quad (39)$$

subject to the integral equation constraint:

$$q(t) = \int_0^t K(t, \tau) u(\tau) d\tau \quad (40)$$

$q(t)$ is a scalar state variable and q_d denotes the desired value of $q(t)$. u_s denotes the steady state value of unbounded scalar controller $u(t)$.

For this problem H is given by

$$H = [u(t) - u_s]^2 + [q_d - q(t)]^2 + \int_t^{t_1} p(\tau) K_\tau(\tau, t) u(t) d\tau \quad (41)$$

The controller u which minimizes H is obtained by setting $\frac{\partial H}{\partial u}$ equal to zero, thus

$$u^*(t) = u_s - 1/2 \int_t^{t_1} p(\tau) K_\tau(\tau, t) d\tau \quad (42)$$

Note that u^* is a functional of p and that H is a function of q . Now

$$\begin{aligned} \dot{p} &= -H_q^* \\ &= 2[q_d - q(t)] \end{aligned} \quad (43)$$

In equation (42), carrying out the integration by parts and substituting for $\dot{p}$ from (43) yields

$$u^*(t) = u_s + \int_t^{t_1} [q_d - q(\tau)] K(\tau, t) d\tau \quad (44)$$

where $p(t_1)$, $t_1 = \infty$, has been set equal to zero to satisfy the transversality condition [3].

Substituting for $q(t)$ from (40) in equation (44) there results

$$u^*(t) = u_s + \int_t^{t_1} q_d K(\tau, t) d\tau - \int_t^{t_1} d\tau \int_0^\tau K(\tau, \sigma) u^*(\sigma) d\sigma \quad (45)$$

Thus, the solution of the above integral equation yields the optimal controller u . For the solution of equation (45) refer to [7]. It may be noted that equation (45) was derived in [7] by using classical variational techniques.

CONCLUSION

Sufficient conditions for optimal control of distributed systems have been given for the case where the control input is a function of time only and where the state variable is to be controlled at a specific point in space. A set of canonical equations for the distributed systems are also given. The results obtained here are compared with those given by Butkovskii in reference [3] and Goodson and Khatri in [7].

ACKNOWLEDGMENT

This work has been supported in part by National Science Foundation Grant No. GK-1323 at Montana State University.

REFERENCES

1. P. K. C. Wang, "Control of Distributed Parameter Systems," A contribution to the book "Advances in Control Systems Theory and Applications," edited by C. T. Leondes, Vol. 1, 1964, Academic Press.
2. K. A. Lure, "The Mayer-Bolza Problem for Multiple Integrals and Optimization of Systems with Distributed Parameters," *Journal of Applied Mathematics and Mechanics*, Vol. 27, No. 5, 1963.
3. A. G. Butkovskii, "Optimal Processes in Distributed Parameter Systems," *Automation and Remote Control*, Vol. 22, No. 1, 1961, pp. 13-21.
4. A. G. Butkovskii, "The Broadened Principle of the Maximum for Optimal Control Problems," *Automation and Remote Control*, Vol. 24, No. 3, 1963, pp. 292-304.
5. Y. Sakawa, "Solution of an Optimal Control Problem in a Distributed Parameter System," *IEEE Transactions on Automatic Control*, Vol. AC-9, October 1964, pp. 420-426.
6. P. K. C. Wang and F. Tung, "Optimal Control of Distributed Parameter Systems," *Journal of Basic Engineering, ASME Transactions, Series D*, Vol. 86, 1964, pp. 67-79.
7. R. E. Goodson and H. C. Khatri, "Optimal Feedback Solutions for a class of Distributed Systems," *Journal of Basic Engineering, ASME Transactions, Series D*, Vol. 88, No. 2, pp. 337-342.
8. A. G. Butkovskii, "Some Approximate Methods for Solving Problems of optimal Control of Distributed Parameter Systems," *Automation and Remote Control*, Vol. 22, No. 10, 1961, pp. 1156-1168.
9. M. M. Denn, R. D. Gray, and J. R. Ferron, "Optimization in a Class of Distributed Parameter Systems," *I & EC Fundamentals*, Vol. 5, February 1966, pp. 59-66.
10. A. I. Egorov, "Optimal Processes in Systems Containing Distributed Parameter Plants I," *Automation and Remote Control*, Vol. 26, No. 6, 1965, pp. 977-994.
11. A. I. Egorov, "Optimal Processes in Systems Containing Distributed Parameter Plants II," *Automation and Remote Control*, Vol. 26, No. 7, 1965, pp. 1188-1196.
12. M. Athans and P. L. Falb, "Optimal Control," McGraw-Hill Book Co., New York, N. Y., 1966, pp. 353-363.
13. R. E. Kalman, "The Theory of Optimal Control and the Calculus of Variations," in the book "Mathematical Optimization Technique" edited by R. Bellman, University of California Press, Berkeley, Calif., 1963.
14. E. W. Hobson, "The Theory of Functions of a Real Variable and the Theory of Fourier's Series," Vol. II, Reissued by Dover Publications, Inc., New York, N. Y., 1957, pp. 338-353.
15. J. Edwards, "The Differential Calculus," Macmillan and Co., Ltd., London, U. K., 1948, pp. 112-125.
16. R. G. Sanger, "Functions of Lines and the Calculus of Variations," in "Contributions to the Calculus of Variations 1931-1932; Theses submitted to the Department of Mathematics of the University of Chicago," The University of Chicago Press, Chicago, Illinois.

V. J. TARASSOV

Western Electric Company, Inc.
Engineering Research Center
Princeton, New Jersey

H. J. PERLIS

Newark College of Engineering
Dept. of Electrical Engineering
Newark, New Jersey

ABSTRACT

The multiparameter optimization is performed on the basis of an integral type of cost functional with a weighting factor on the control effort. The distributed system dynamics are assumed to be governed by a set of coupled first-order partial differential equations with the same principal part. The maximum condition is derived by means of the perturbation of the cost functional. The system theory is illustrated by a numerical example dealing with water pollution. The optimal control of dissolved oxygen in a stream is determined under realistic aquatic environmental conditions. The resultant control policy can be instrumented in a direct digital control system.

INTRODUCTION

This paper is concerned with the optimization of quasi-linear multi-variable distributed parameter systems. The processes studied involve multiple dependent variables which are functions of both the temporal and spatial independent variables. The optimization is developed for a general integral type of cost functional with weighted energy constraints.

The admissible system dynamics is restricted to be of the coupled quasi-linear first-order partial differential equation type. This method of analysis is particularly useful for systems of partial differential equations with the same principal part. Under this condition the actual number of integrations is reduced when the integrations are performed along the characteristic directions. This type of problem is frequently encountered in process control applications dealing with plug flow tubular reactors. Similar applications can be found in distributed electrical networks.

The maximum principle is formulated and proved for this class of systems. This constitutes a theoretical extension of Sirazetdinov's work to the multiple dependent variable class of systems [1]. The perturbation of the cost functional from its optimal solution is examined; this leads to an increase in the cost functional. This variational operation generates the costate system of equations with the associated boundary conditions, and the necessary conditions for optimality are defined.

The minimum for the cost functional is obtained by means of an iterative technique. A weighted gradient of the generalized Hamiltonian for the constrained system is used to provide successive improvements for the approximation of the optimal solution.

The theory is illustrated by means of a one-dimensional example. The very practical problem of water pollution control was selected for this purpose. There are two dependent variables in this problem: biochemical oxygen demand (BOD) and dissolved oxygen (DO), and there are two independent variables: space and time. The mathematical model for the stream is an extension of the Streeter-Phelps equations.

Solutions to this problem were generated with the use of a digital computer. Resulting optimal control, state variable, and costate variable surfaces are presented. Employing typical hydrological coefficients in the problem, one can see that the optimal control surface lies within the operational regime of the current state-of-the-art in aeration equipment. The paper concludes with a discussion of an implementation procedure of the optimal control policy in a direct digital control loop.

Restrictions on Admissible Systems

The optimal control processes whose dynamics are governed by sets of multivariable systems of quasi-linear partial differential equations of the first order will be considered here. Specifically, the process dynamics are assumed to be of the following form:

$$\frac{\partial v_i}{\partial t} = f_i(t, x, v, v_q, u) \quad (1)$$

where:

$$x \equiv (x_1, x_2, \dots, x_n) \quad v_q \equiv \frac{\partial v_i}{\partial x_j} \text{ for } \left\{ \begin{array}{l} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{array} \right\}$$

$$v \equiv (v_1, v_2, \dots, v_m)$$

$$u \equiv (u_1, u_2, \dots, u_l)$$

the functions " f_i " for $(i = 1, 2, \dots, m)$ are restricted to the following quasi-linear form:

$$f_i = f_i^0(t, x, v, u) + \sum_{j=1}^n f_i^j(t, x, u) \frac{\partial v_i}{\partial x_j} \quad (2)$$

the functions " f_i " are defined over the space $[t_0, t_f] \times [x^0, x^f] \equiv [TxX]$ and are assumed to possess continuous second order derivatives with respect to the independent variables.

The set of admissible control variables $u_k(t, x)$ are assumed to satisfy the following functional constraints:

$$g(u_1(t, x), u_2(t, x), \dots, u_l(t, x)) \leq 0 \quad (3)$$

The system's initial and boundary behavior is specified in the following manner:

$$v_i(0, x) = \phi_i(x); v_i(t, 0) = \psi_i(t) \quad (4)$$

Restriction on System Performance Criteria

It is assumed that the system's merit is evaluated by means of a cost functional of the following form:

$$J = \int_{t_0}^{t_f} \int \mathcal{J}(t, x, v, v_q, u) dt dx \quad (5)$$

where:

$$\mathcal{J} = \mathcal{J}_0(t, x, v, u) + \sum_{i=1}^m \sum_{j=1}^n G_{ij}(t, x, u) \frac{\partial v_i}{\partial x_j} \quad (6)$$

Formulation of the Maximum Principle

The optimal control problem is to find a control policy which will minimize the cost functional given by equation (5). The starting point for the establishment of a maximum principle is the formation of a generalized Hamiltonian " H " with appropriate constraints. This is given by the following:

$$H = H(t, x, v, v_q, p, u) = -\mathcal{J} + \langle p_i, f_i \rangle \quad (7)$$

where:

$\mathcal{J}$: is the cost functional to be minimized (therefore " H " must be maximized)

f_i : represents the dynamic constraints of the system

p_i : represents the costate vectors

$\langle, \rangle$: represents the sum of the inner products of the specified vectors.

The controlling functions $u^*(t, x)$ shall satisfy the maximum principle, if for any arbitrary point $\zeta \in TxX$, the function $H(t, x, v, v_q, p, u)$ attains its

absolute maximum on the set of admissible controlling functions $u(t,x)$. This condition is alternately expressed by:

$$\Delta H = H(t,x,v,v_q,p,u^*(t,x) + \Delta u) - H(t,x,v,v_q,p,u^*(t,x)) < 0 \quad (8)$$

Theorem

If the controlling functions $u^*(t,x)$ ensure the minimum of the functional expressed in equation (5), then they satisfy the maximum principle.

The summary of the proof of this theorem is found in the Appendix.

Summary of Results Developed in the Proof of the Necessary Conditions for Optimality

The following equations are derived in the Appendix. They represent the necessary conditions that must be imposed on the costate system of equations to insure the maximum of the generalized Hamiltonian "H". The costate system dynamics are defined by the following two equations:

$$\frac{\partial v_i}{\partial t} = \frac{\partial H}{\partial p_i}; \quad \frac{\partial p_i}{\partial t} = -\frac{\partial H}{\partial v_i} + \sum_{j=1}^n \frac{\partial}{\partial x_j} \left(\frac{\partial H}{\partial v_{qj}} \right) \quad (9)$$

The terminal and boundary conditions for the costate system of equations are given by:

$$\sum_{j=1}^n \frac{\partial H}{\partial v_{ix_j}} \cos(N, x) \Big|_{x^f} = 0; \quad p_i(T, x) = 0 \quad (10)$$

for $i = 1, 2, \dots, m$. N represents the outward normal of the terminal hypersurface.

ILLUSTRATION OF THEORY BY PHYSICAL PROBLEM

Formulation of Problem

The specific objective of the optimal control policy is the maintenance of dissolved oxygen of the stream at or near a prescribed level by means of artificial aeration elements, and, at the same time, minimizing the exerted control effort. The process dynamics are governed by the following two state equations [2]:

$$\frac{\partial v_1}{\partial t} = -K_R(t,x)v_1 - v(t,x) \frac{\partial v_1}{\partial x} \quad (11)$$

$$\frac{\partial v_2}{\partial t} = -K_a(t,x)v_2 - K_d(t,x)v_1 + K_3(t,x) + u(t,x) - v(t,x) \frac{\partial v_2}{\partial x} \quad (12)$$

The various factors are defined as follows:

$v_1 \equiv$ biological oxygen demand (BOD)

$v_2 \equiv$ dissolved oxygen (DO)

$K_R \equiv$ BOD removal coefficient, DAY^{-1}

$K_a \equiv$ reaeration coefficient, DAY^{-1}

$K_d \equiv$ deoxygenation coefficient, DAY^{-1}

$K_3 \equiv$ natural source of DO, $\text{mg./l.} \times \text{DAY}$

$u \equiv$ control variable, $\text{mg./l.} \times \text{DAY}$

$v_2 \equiv$ velocity of stream, mi./DAY

The specific values for the various parameters are taken from Davidson's publication [3]:

$$\begin{aligned}K_r &= 0.16 \text{ l/D} & K_3 &= 6.0 \text{ mg./l.} \\K_a &= 0.66 \text{ l/D} & K_d &= 0.16 \text{ l/D} \\v_1(0) &= 30 \text{ mg./l.} & v &= 1.0 \text{ M/D} \\v_2(0) &= 6 \text{ mg./l.} & L &= 5 \text{ M}\end{aligned}$$

A quadratic cost functional has been selected for this illustration. It is of the following form:

$$J = \iint (v_{sp} - v_2)^2 dt dx + k \iint u^2 dt dx \quad (13)$$

where v_{sp} is the specified DO profile ($v_{sp} = 6.0$) and k is the control weighting factor ($k = 0.5$).

Application of the Maximum Principle

The maximum principle is applied to the system described by equation (11) and equation (12). When the form for the cost functional is assumed to be given by equation (13), then, the costate vectors for this process become:

$$\frac{\partial p_1}{\partial t} = K_d p_1 + K_a p_2 - v \frac{\partial p_1}{\partial x} \quad (14)$$

$$\frac{\partial p_2}{\partial t} = -2(v_{sp} - v_2) + K_a p_2 - v \frac{\partial p_2}{\partial x} \quad (15)$$

with the following terminal and boundary conditions:

$$p_1(T, x) = p_2(T, x) = p_1(t, L) = p_2(t, L) = 0$$

This, then, completes the derivation of the necessary equations for obtaining the optimal solution to the posed problem.

Gradient Search for Maximum of "H"

Since the cost functional is constructed to be convex, the gradient search technique can be used to establish the local maximum of H (and consequently the minimum of J). When the system is not under the influence of the optimal control, the "direction" toward the optimal control is dictated by the positiveness of the gradient of the Hamiltonian functional:

$$\iint \left(\frac{\partial H}{\partial u_k}, \Delta u_k \right) dt dx > 0 \quad (16)$$

where:

$$\Delta u_k = u_k^{(n+1)}(t, x) - u_k^{(n)}(t, x) \quad (17)$$

The superscript on the control variables indicate the iteration number. This iteration technique eventually yields the optimal control to a desired accuracy.

Numerical Integration of Partial Differential Equation

Since the partial differential equations encountered in this work have the same principal part, the method of characteristics can be used to integrate the system of equations^[4]. The standard fourth order Runge-Kutta process is used to integrate the resultant equations along the characteristic directions.

Presentation of Results

The optimal solution for this problem was obtained in seven iterations. This resulted in the value 26.90 for the cost functional. The resultant state, costate, and optimal control surfaces are presented in figures 1-5 respectively. All the drawings have been normalized with respect to the

maximum amplitude which has been set equal to two inches. The maximum amplitudes for the five figures are in order as follows: 30.00, 6.0, 0.63, 1.7, 1.6. The resultant optimal control surface lies within the operational region of the current state-of-the-art in aeration equipment.

Implementation of Optimal Control

The configuration shown in Figure 6 can be used to implement the optimal closed loop system for the class of processes that are considered in this work. The greatest difficulty lies in implementing the distributed measurement.

SUMMARY AND CONCLUSIONS

The results presented here are both of the theoretical and applied nature. The theoretical part is formed by the extension of optimal control theory to multivariable systems with quasi-linear partial differential equations. The applied nature is presented in the numerical solution of the optimal control problem and the implementation of the resultant control policy.

APPENDIX

Determination of the Necessary Conditions for Optimality

If there exists an admissible optimal control policy which yields a minimum for the cost functional J , then, any different admissible control policy will produce a non-decreasing cost functional.

The cost functional is expressed in terms of the generalized Hamiltonian functional and the dynamic constraint is as follows:

$$J = \iint (-H + \langle p_i, \frac{\partial v_i}{\partial t} \rangle) dt d\tau \quad (18)$$

Now, a small perturbation of the control variable will result in the following change in the cost functional:

$$\Delta J = \iint \{ \langle p_i, \frac{\partial \Delta v_i}{\partial t} \rangle + \langle \Delta p_i, \frac{\partial v_i}{\partial t} \rangle + \langle \Delta p_i, \frac{\partial v_i}{\partial t} \rangle + \Delta H^* \} dt d\tau \quad (19)$$

where:

$$\Delta H^* = \Delta H + \sum_{i=1}^m \frac{\partial H}{\partial v_i} \Delta v_i + \sum_{q \neq v_q} \frac{\partial H}{\partial v_q} \Delta v_q + \sum_{i=1}^m \frac{\partial H}{\partial p_i} \Delta p_i + \epsilon \quad (20)$$

$$\Delta H = H(t, x, v, v_q, p, u + \Delta u) - H(t, x, v, v_q, p, u) \quad (21)$$

ϵ contains the second order terms.

The following equation is obtained by expanding and regrouping the above factors [5]:

$$\begin{aligned} \Delta J = \iint \{ & -\Delta H + \langle \left[\frac{\partial v_i}{\partial t} - \frac{\partial H}{\partial p_i} \right], \Delta p_i \rangle + \langle \left[-\frac{\partial p_i}{\partial t} - \frac{\partial H}{\partial v_i} \right. \\ & + \sum_{j=1}^n \frac{\partial}{\partial x_j} \left(\frac{\partial H}{\partial v_q} \right)], \Delta v_i \rangle + \frac{\partial}{\partial t} \langle \Delta p_i, \Delta v_i \rangle \rangle \\ & + \sum_{j=1}^n \sum_q \frac{\partial}{\partial x_j} \left(\frac{\partial H}{\partial v_q} \Delta v_i \right) + \langle \Delta p_i, \frac{\partial \Delta v_i}{\partial t} \rangle - \beta \} dt d\tau \end{aligned} \quad (22)$$

The costate system of equations is obtained from the equation (22). They are:

$$\frac{\partial v_i}{\partial t} = \frac{\partial H}{\partial p_i}; \quad \frac{\partial p_i}{\partial t} = -\frac{\partial H}{\partial v_i} + \sum_{j=1}^n \frac{\partial^2 H}{\partial x_j \partial v_q} \quad (23)$$

with the following boundary and terminal conditions:

$$\sum_{j=1}^n \frac{\partial H}{\partial v_{i x_j}} \cos(N, X) \Big|_{x^f} = 0; \quad p_i(T, X) = 0 \quad (24)$$

for $i = 1, 2, \dots, m$,

where N is the outward normal of the terminal surface.

The remainder β is shown to be of the following form for any arbitrary point in the closed bounded state space:

$$\beta \leq c \Delta u^2 \Delta t^2$$

where: c is an arbitrary finite control

Δu represents the largest perturbation of the control variable.

Δt represents the increment of time during the perturbation.

Summary of the Proof of the Maximum Principle

The maximum principle states that the cost functional J is minimized where H assumes a maximum (or equivalently $\Delta H \leq 0$) on the set of admissible control variables. The proof of the principle is established by contradiction.

Select any point $(t, x_i) \in [T, X]$ and assume that there exists an optimal control $u^*(t, x)$ such that $\Delta H > 0$, then, in view of the continuities assumed in the formation of the Hamiltonian, there must exist some control $u^0(t, x)$ in the neighborhood of $u^*(t, x)$ for which the following condition will hold:

$$\Delta H > 0 \quad (25)$$

The neighborhood is defined as $\Delta \sigma \epsilon \Delta t \times \Delta \tau$ where $\Delta t = t_2 - t_1$, and $\Delta x_k \epsilon \Delta \tau$. Consider the following composite function $u^0(t, x)$, which is equal to $u^*(t, x)$ outside of the region $\Delta \sigma$, and which is equal to $u^0(t, x)$ inside the region $\Delta \sigma$. Then, the equation becomes:

$$\Delta J = - \iint \alpha \left[1 - \frac{c \Delta u^2 \Delta t^2}{\alpha \Delta \tau} \right] dt d\tau \quad (26)$$

Since the subspace $\Delta \tau$ is assumed to be finite, $\Delta \tau \neq 0$, therefore, by selecting Δt sufficiently small, the contribution of the second term under the integral sign can be made insignificant when compared to unity. Therefore, the argument under the integral sign will be positive. This implies that there is a control, other than the optimal, for which $\Delta J < 0$. This, however, contradicts the definition of the existence of the optimal control. Thus the condition $\Delta H \leq 0$ is a necessary condition for optimality.

ACKNOWLEDGEMENT

The authors would like to acknowledge National Science Foundation (Grant GK-1041), Department of Interior, Office of Water Resources Research (Grant A-017 N. J.), and the Foundation for the Advancement of Graduate Study in Engineering for their support. The generous grant of

computer time by Rutgers University Computer Center is also appreciated.

REFERENCES

- (1) Sirazetdinov, T. K., "On the Theory of Optimal Processes with Distributed Parameters," Automation and Remote Control, Vol. 25, No. 4, 1964.
- (2) O'Conner, D. J., "Proceedings of the Eleventh Summer Institute in Water Pollution Control," Manhattan College, June, 1966.
- (3) Davidson, B., and Bradshaw, R. W., "Thermal Pollution of Water Systems," Environmental Science and Technology, pp. 618-630, August, 1967.
- (4) Courent, R., and Hilbert, D., "Methods of Mathematical Physics," Vol. II, Interscience Publication, New York, 1962.
- (5) Tarasov, V. J., "Optimization of Multivariable Distributed Parameter Systems," Ph.D. Thesis, Rutgers University, New Brunswick, N. J., October, 1968.

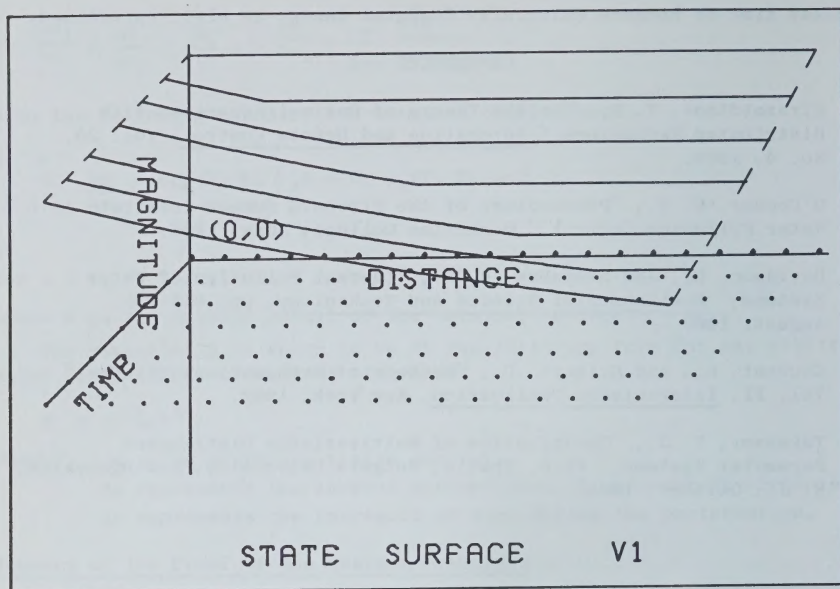


FIG. 1

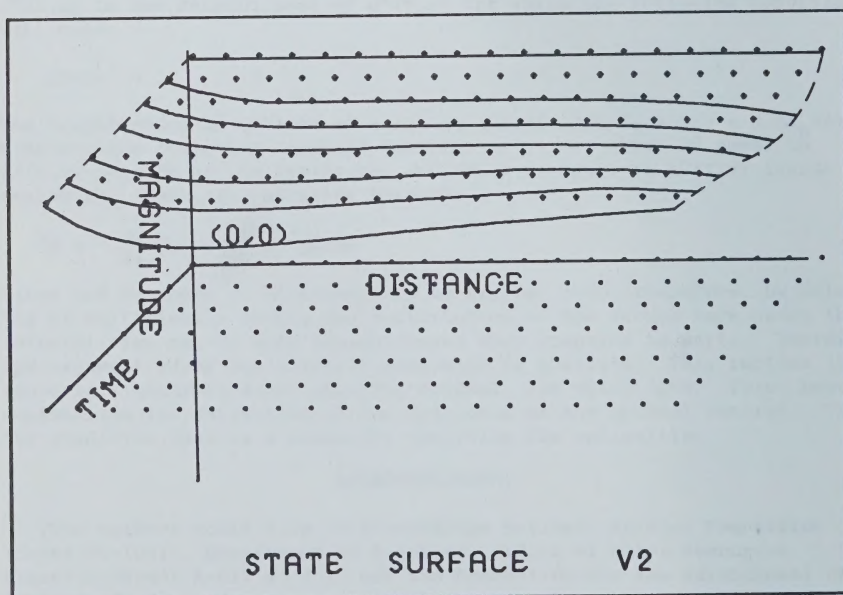


FIG. 2

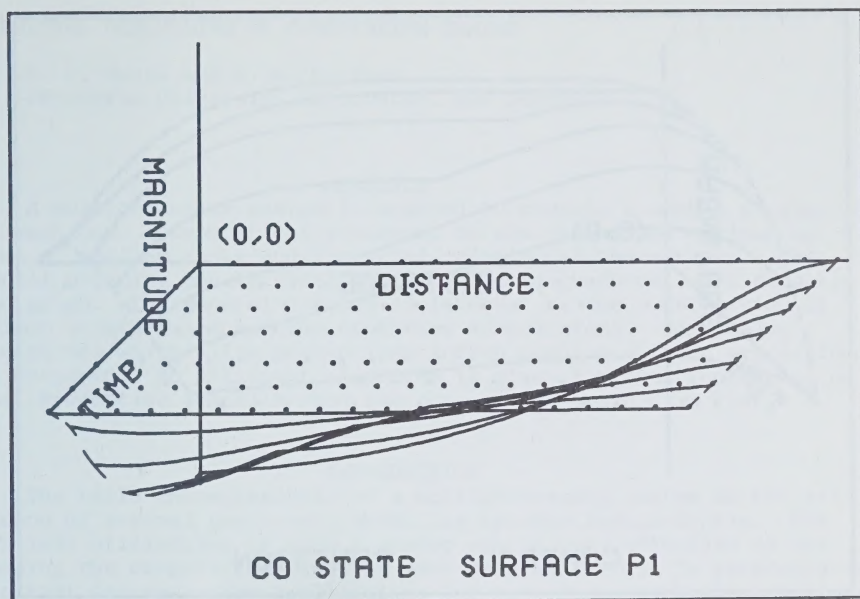


FIG. 3

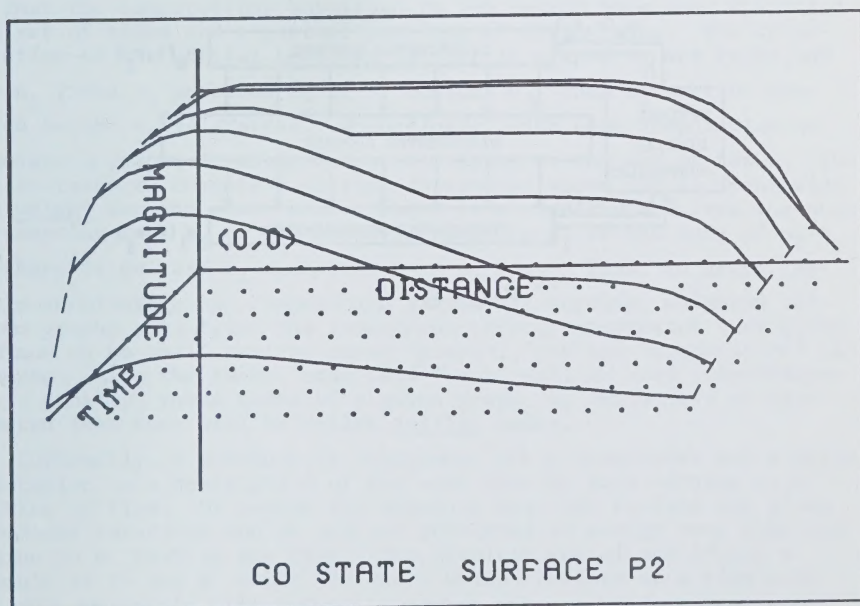


FIG. 4

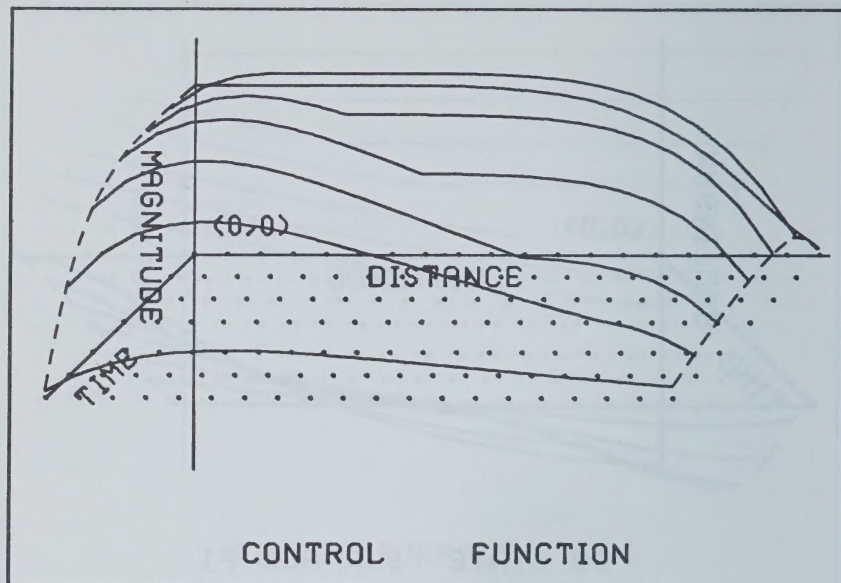


FIG. 5

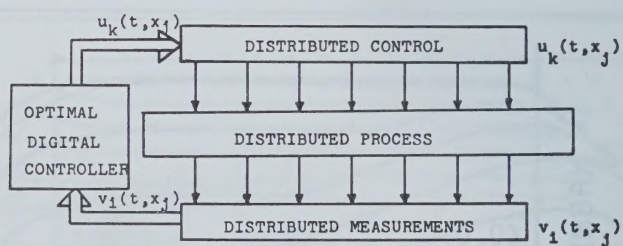


FIG. 6

PREEMPTIVE SCHEDULING OF COMPUTATION GRAPHS

R. R. Muntz and E. G. Coffman
Princeton University, Princeton, New Jersey

ABSTRACT

A multiprocessor system is assumed to contain a number of identical machines. Computations presented to the system are defined in terms of smaller tasks and a partial ordering of these tasks. This partial ordering can be specified by a linear graph called a computation graph. The general problem of interest is the construction of minimum length schedules for arbitrary computations. Scheduling disciplines which allow preemptions and/or machine sharing are defined and compared. An efficient algorithm is given for constructing an optimal preemptive schedule when the computation graph is a tree.

1. INTRODUCTION

The basic characteristic of a multiprocessing system is the existence of several processors which can operate independently. The efficient utilization of such a system can be very effective in decreasing the computation times of many programs. This is particularly important for real-time applications but also for lengthy computations [1], such as weather forecasting [2], when the results are needed more quickly than they can be provided by a single processor. In this paper we report results from some investigations into the problem of making optimum use of a multiprocessing system in decreasing computation times.

We shall assume that all processors in the system are identical and that the computations submitted to the system have been specified as a set of tasks and a partial ordering of these tasks. The interpretation of the partial ordering is that if n_i and n_j are tasks and $n_i > n_j$ (read n_i precedes n_j or n_j follows n_i) then n_i must be completed before n_j is started. Accordingly, with each computation we associate a graph, G , whose nodes correspond to the set of tasks. The time it takes to compute each task is assumed known and is identified as a weight for the given node. There is a directed arc from the node representing task n_i to the node representing n_j if and only if $n_i > n_j$ and there is no task n_k such that $n_i > n_k > n_j$. Thus, in graph theoretic terminology our computation graphs are acyclic, weighted, directed graphs satisfying the (non-transitivity) constraint just given. From now on we shall use the terms "graph", "job" and "computation" as synonyms. Also the terms "node" and "task" will be used interchangeably. Finally, those nodes of a given graph, G , which have no arcs directed into them will be called initial nodes.

Informally, a schedule or assignment for k processors and a given computation is a description of the work done by each machine as a function of time. Of course the schedule must not violate the given precedence relations and we are not permitted to assign more than one machine to a task at any time. The simplest way of specifying a schedule is to use a Gantt [3] Chart which consists of a time axis for each processor with intervals marked off and labeled with the name of the task being computed. We use the symbol θ to represent

an idle period.

We are interested in the problem of constructing minimum length assignments. In most investigations of this problem it is assumed that once a processor is assigned to a task it must work continuously on this task until it has been completed. This we term the Basic Assignment (BA) discipline. Clearly, the problem of finding an optimal BA for any given graph is effectively solvable by exhaustion. However, a practical solution is known only for the case when G is a tree with equally weighted nodes [5].

We shall consider the effect of replacing the BA condition to allow processors to be interrupted before a task is completed and re-assigned to a new task. We call this the Preemptive Assignment (PA) discipline. The potential benefits derivable from preemptions can be seen by the example shown in Figure 1. The schedule shown in Figure 1(b) is a minimum length assignment for G with $k = 2$ machines using the BA discipline. In Figure 1(c) we compute G in 1.5 time units with one preemption. We have assumed that there is no cost associated with preemptions. We will continue to make this assumption throughout this paper.

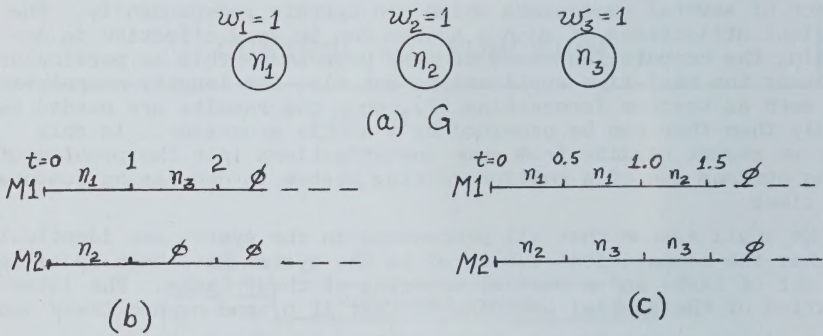


FIGURE 1

There is another variation of the BA discipline that we want to discuss. Consider that the k processors in a system comprise a certain amount of computing capability rather than being discrete units. Assume further that this computing capability can be assigned to tasks in any amount between 0 and the equivalent of one machine. If we assign α , $0 < \alpha \leq 1$ computing capability to a task then we assume that the computation time of the task is increased by a factor of $1/\alpha$. For example, if we assign $1/2$ of a machine to task n_1 , which has weight w_1 , then it will take $2w_1$ units of time to complete n_1 . Up to this point we have not allowed the amount of computing capability assigned to a task to be changed until the task is completed. We call this the Pure Time-Sharing Assignment (PTSA) discipline. If we allow the amount of computing capability assigned to a task to change before the task is completed (we include the case where the task is not worked on at all for some interval) then we call this the General Assignment (GA) discipline. It should be clear that in allowing the value of α for a given node to change before the completion of the node we have effectively introduced the preemptive capability into the GA discipline.

We do not try to justify the two assignment disciplines we have

just defined in the sense of finding practical situations to which they correspond.* We introduce PTSA primarily for its theoretical interest; however, the GA discipline will be extremely useful in studying preemptive assignment techniques in later sections.

Our first concern will be to define a meaningful method of comparing assignment disciplines and to apply this definition to the disciplines we have discussed above. Following this we shall give an algorithm for constructing an optimal GA for any computation whose graph is a tree; i.e. a graph each of whose nodes has exactly one arc directed away from it, except for a unique root node which has no arc directed away from it.

II. AN ASSIGNMENT DISCIPLINE HIERARCHY

Let D_1 and D_2 be two scheduling disciplines. Then we say that D_1 is at least as powerful as D_2 ($D_1 \geq D_2$ if for any job and any number of machines a schedule conforming to D_1 exists which is at least as short as any schedule conforming to D_2). We say D_1 is strictly more powerful than D_2 ($D_1 > D_2$) if $D_1 \geq D_2$ and there is at least one job and number of machines for which a schedule conforming to D_1 exists which is shorter than any schedule conforming to D_2 . We use the notation $D_1 \equiv D_2$ to mean $D_1 \geq D_2$ and $D_2 \geq D_1$.

In terms of these definitions we shall establish a partial ordering of the four assignment disciplines we have introduced. This partial ordering is illustrated in Figure 2 where an arc from D_1 to D_2 indicates $D_1 > D_2$.

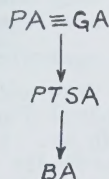


FIGURE 2

We will need the following terminology. If G is a graph and D_1 is an assignment discipline then the minimum computation time of G with k machines and using D_1 is denoted $CT_{D_1}(G, k)$. We proceed now with a sequence of theorems which verify the relations shown in Figure 2.

*Probability models of disciplines very similar to the PTSA discipline have been studied by Coffman [6] and Kleinrock [7] in connection with time-sharing systems.

Theorem 1 - $PTSA > BA$

Proof: We know from their definitions that $PTSA \geq BA$. To complete the proof we give an example of a job and a number of machines for which the minimum computation time using $PTSA$ is less than that using BA .

With $k = 2$ the graph, G , of three independent, equally weighted nodes shown in Figure 1(a) will suffice for this purpose. Clearly $CT_{BA}(G, 2) = 2$. With $PTSA$ we can assign $2/3$ of a machine to each node and complete G in 1.5 units of time. Thus $CT_{PTSA}(G, 2) < CT_{BA}(G, 2)$.

Q.E.D.

Theorem 2 - $PA \equiv GA$. Note that this theorem implies that if preemption is possible then the time-sharing capability is not needed for an optimal assignment.

Proof: Rather than giving a formal proof of this theorem a short intuitive argument is presented. We already know from their definitions that $GA \geq PA$, so if we can show that $PA \geq GA$ we are done. Short of actually proving this statement we illustrate how to approximate the fractional assignment of machines in a GA by preemptions alone. For example, let half a machine be assigned to a node n_i and half a machine be assigned to a node n_j over the interval (t_1, t_2) . We approximate this situation by dividing the interval into $2n$ segments of length $t_2 - t_1/n$ each and assigning the machine to n_i and n_j in alternate segments. As n becomes large it should be clear that this has the same effect, over any finite interval, as the fractional assignment of the machine. Of course this is only true because there is no cost associated with preemptions.

Q.E.D.

The formal proof of this theorem consists of an effective procedure for constructing a PA from a GA for any graph and number of machines. It follows that if we know how to construct an optimal GA for some G and k then we also know how to construct an optimal PA .

Theorem 3 - $PA > PTSA$ (or equivalently $GA > PTSA$)

Proof: We know that $GA \geq PTSA$ and $PA \equiv GA$. It follows that the proof is complete if we can exhibit a graph G and a number of machines k for which $CT_{PA}(G, k) < CT_{PTSA}(G, k)$. Such a situation is illustrated in Figure 3. It is left to the reader to verify that this graph can be computed in 4 units of time with the PA discipline and that a greater time is necessary with the $PTSA$ discipline.

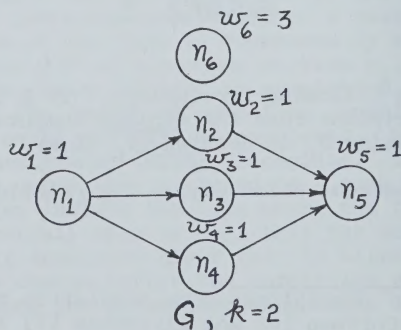


FIGURE 3

III. AN OPTIMAL GA (PA) SCHEDULING ALGORITHM FOR COMPUTATION TREES

We shall now describe an algorithm for constructing an optimal GA (by the remark following Theorem 2 this is equivalent to an algorithm for constructing an optimal PA) for any number of machines when the computation graph is a tree and the node weights are mutually commensurable (i.e. there exists a real number, c , such that cw_i is an integer for all weights, w_i). This last requirement is not very restrictive since we can still approximate node weights arbitrarily closely. Before we state the algorithm and prove its optimality we digress to introduce some notions which we will find useful.

Let G be a graph and $\{w_1, \dots, w_n\}$ be the weights of the nodes of G . Assume the w_i are mutually commensurable and let c be a real number such that cw_i is an integer for all w_i . Let $w = (\frac{1}{c}) \text{g.c.d. } \{cw_i\}$; that is, let cw be the greatest common divisor of the integers $\{cw_i\}$. We now define a graph $G_{w/n}$ where n is any integer, which is derived from G .

Informally, we take each node, n_i , of G with weight w_i and replace it by a series connection of nodes of weight w/n so that the total weight of these nodes equals w_i . (Note that w is indeed the largest unit in multiples of which all node weights can be expressed.) We require that all edges that were incident into n_i are incident into the first node in the series connection. Similarly, all edges that were incident out of n_i now leave the last node in the series connection.

Example. Consider the graph shown in Figure 4(a) for which $c = 1$ and $w = 1$. If we choose $n = 2$ we have the graph, $G_{1/2}$ shown in Figure 4(b).

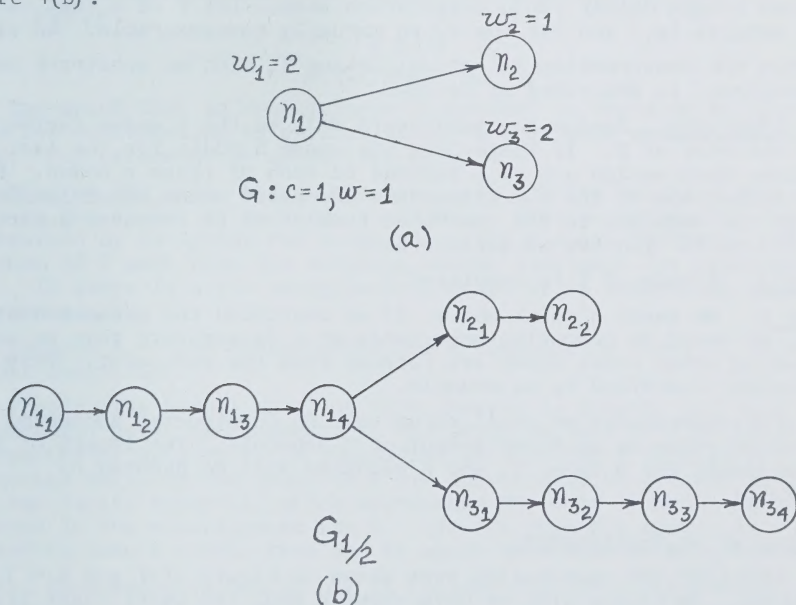


FIGURE 4

A BA for $G_{w/n}$ is also a PA for G in the obvious sense. In fact, instead of referring to a BA for $G_{w/n}$ we could talk about a PA for G in which preemptions are only allowed at times equal to some multiple of w/n . It should be clear that as n approaches infinity the effect of this restriction diminishes and we can approximate arbitrarily closely any PA for G by a BA for $G_{w/n}$. The following lemma is a direct consequence of this fact and is stated without proof. This lemma will be exploited in the discussion of the GA algorithm for trees.

Lemma 1 - Let G , w_i , c , and w be as defined above and assume there are k available machines. Then we can find a constant K such that

$$CT_{PA}(G, k) \leq CT_{BA}(G_{w/n}, k) \leq CT_{PA}(G, k) + \frac{K}{n} \quad n = 1, 2, \dots$$

Before we proceed it will be helpful to introduce the following terminology connected with trees. Consider a computation tree, T , and two nodes $n_i, n_j \in N(T)$ such that $n_i < n_j$. The distance between n_i and n_j is the sum of the node weights for each node (including both n_i and n_j) in the (unique) path connecting n_i and n_j . If n_i is the root of T then this distance is called the height of the node n_j . Finally, at any point in the execution of a schedule for a tree the available nodes are the initial nodes of the tree remaining to be processed. These are the (independent) nodes that may be processed without violating any precedence constraints.

We return now to a statement of the algorithm for obtaining a minimal length GA(PA) for a computation tree. Let T be a tree with node weights $\{w_i\}$ and let the w_i be mutually commensurable. An algorithm for constructing an optimal GA for T , with an arbitrary number of machines, is described as follows.

Algorithm. Assign one machine ($\alpha = 1$) to the k nodes farthest from the root of T . If there is a tie among b nodes for the last a machines then assign a/b of a machine to each of these b nodes. Each time either one of the two situations described below occurs we reassign the machines to the tree that remains to be computed according to this rule. The two situations are:

Event 1. A node of T is completed.

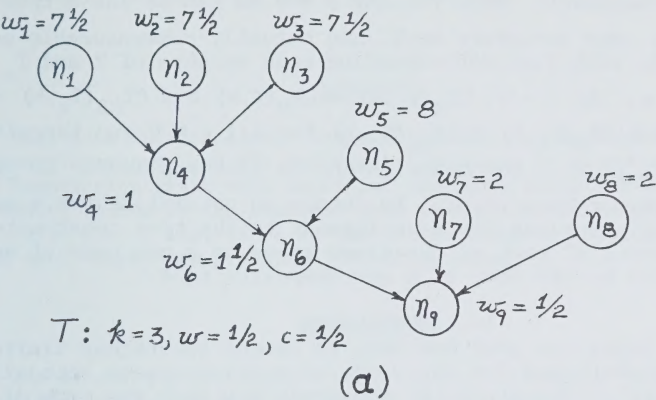
Event 2. We reach a point where, if we continued the present assignment, we would be computing some nodes at a faster rate than we are computing other nodes which are farther from the root of t . This will be further clarified by an example.

For convenience we shall refer to a GA constructed according to the above rules as an M-assignment or M-schedule. The length of an M-assignment for a tree, T , and k machines will be denoted by $CT_M(T, k)$.

Example of an M-assignment

Consider the computation tree shown in Figure 5(a) and $k = 3$ machines. In Figure 5(b) we have shown a modified Gantt chart illustrating the M-assignment for T . Observe that at $t = 1/2$ there is an occurrence of Event 2 because the heights of nodes n_1, n_2 , and n_3 have been reduced (prior to completion) to the height of n_5 (viz. $1/2$).

At this point the three processors are shared equally by the nodes n_1, n_2, n_3 and n_5 . Incidentally the minimum computation time of T using the BA discipline is $17\frac{1}{2}$.



	$t=0$	$t=1/2$	$9\frac{5}{6}$	$10\frac{5}{6}$	$12\frac{1}{3}$	$12\frac{5}{6}$	
M1	n_1	n_1	n_4	n_6	n_9		
M2	n_2	n_2	n_7	n_7	ϕ		ϕ
M3	n_3	n_3	n_5	n_8	ϕ		
EVENT	2		1	1	1	1	

(b)

FIGURE 5

The proof that an M-assignment is optimal is based on an algorithm for constructing an optimal BA for a tree with equally weighted nodes. For reference we state this result.

Theorem (Hu [5]) - Let T be a tree with equally weighted nodes. Let k be the number of machines available. Then an optimal BA for T is obtained by assigning the machines to the k nodes farthest from the root of T each time the machines become available for reassignment. If there is a tie among several nodes it is immaterial which are chosen.

Theorem 4 - The M-assignment for a computation tree, T, is a minimal length GA.

Proof: The formal proof of this theorem is lengthy and will not be presented here. The essence of the proof is that if a BA for $T_{w/n}$ constructed by Hu's rule, is considered as a PA for T then as n approaches infinity the nodes of T are computed at an average rate, over any finite interval, which approaches the rate at which they are computed in the M-assignment for T. (To see this the reader is invited to take a simple tree and to apply Hu's rule to $T_{w/n}$ for several increasing values of n.) It follows that as n approaches infinity $CT_{BA}(T_{w/n}, k)$ approaches $CT_M(T, k)$. But by Lemma 1 we know that as $n \rightarrow \infty$, $CT_{BA}(T_{w/n}, k) \rightarrow CT_{PA}(T, k)$ and therefore $CT_M(T, k) = CT_{PA}(T, k)$. Q.E.D.

It should be pointed out that the algorithm we have presented is optimal even if the node weights are not mutually commensurable. Let T be a tree with arbitrarily weighted nodes and let k be the number of available machines. Then for any $\epsilon > 0$ we can define a tree T_ϵ which has the same structure as T , has mutually commensurable node weights and is such that corresponding node weights of T and T_ϵ differ by less than ϵ . As $\epsilon \rightarrow 0$, $CT_M(T_\epsilon, k) \rightarrow CT_M(T, k)$ and $CT_{GA}(T_\epsilon, k) \rightarrow CT_{GA}(T, k)$. But $CT_M(T_\epsilon, k) = CT_{GA}(T_\epsilon, k)$ for all $\epsilon > 0$ and therefore $CT_M(T, k) = CT_{GA}(T, k)$. The M-assignment is also applicable to forests (or sets) of computation trees. To obtain an optimal GA for a set of trees one simply applies the M-assignment to the tree constructed by joining the roots of each of the given trees to a new node of weight zero which will be the root of a new composite tree.

IV. CONCLUSION

In this paper our goal has been to extend the rather limited theory so far developed for the study of multiprocessor scheduling. The major point of departure for our study has been the work of Hu in BA scheduling of computation trees. The principal generalizations introduced are those of preemptive (PA) and time-sharing (PTSA) assignments in which computations are allowed to be interrupted and shelved in favor of other tasks whose earlier computation leads to greater efficiency, and in which machines may be shared by more than one task at a cost in computation rate for each task. The basic results have been

- (1) a proof that for general graphs PA assignments can always be made at least as short as PTSA assignments.
- (2) a proof that for general graphs scheduling with preemptions is not improved by allowing processor sharing, and
- (3) the statement and proof of an optimal GA algorithm for computation trees.

REFERENCES

- [1] Estrin G., "The Organization of Computer Systems--The Fixed Plus Variable Structure Computer," Proc. WJCC, Vol.17, May 1960, pp. 33-40.
- [2] Martin, David, "The Automatic Assignment and Sequencing of Computations on Parallel Processor Systems," Doctoral Dissertation, University of California at Los Angeles, 1966.
- [3] Heller, J., "Sequencing Aspects of Multiprogramming," Journal of the ACM, Vol. 8, No. 3, July 1961, pp. 426-439.
- [4] McNaughton, R., "Scheduling with Deadlines and Loss Functions," Management Science, Vol. 6, No. 1, October 1959.
- [5] Hu, T.C., "Parallel Sequencing and Assembly Line Problems," Operations Research, Vol. 9, No. 6, November 1961.
- [6] Coffman, E.G. and L. Kleinrock, "Feedback Queueing Models for Time-Shared Systems," Journal of the ACM (to appear October 1968).
- [7] Kleinrock, L., "Time-Shared Systems: A Theoretical Treatment," Journal of the ACM, Vol. 14, No. 2, April 1967, pp. 242-261.

BOUNDS ON MEMORY REQUIREMENTS OF MULTIPROCESSING SYSTEMS

Y-C. E. CHEN and D. L. EPLEY

University of Iowa, Iowa City, Iowa 52240

ABSTRACT

Real-time multiprocessor computer systems are modeled by a multi-level memory structure and the operational software by computation graphs. Given a computation graph with specified execution times and main memory requirements, bounds on the required main memory and the inter-memory channel rates are of importance to the system designer. Algorithms for deriving such bounds are presented and illustrated.

INTRODUCTION

The scheduling of realtime tasks on multiprocessor computer systems is a significant problem in the design of system operating programs for such systems. Using the simple hardware model shown in Figure 1(a) and a computation graphs to represent interdependence of realtime tasks, the scheduling problem has been studied by several authors. (Ref. 1-6) Using these prior scheduling methods and the more realistic model of Fig. 1(b), bounds on the number of processors, main memory size and channel speed will be investigated.

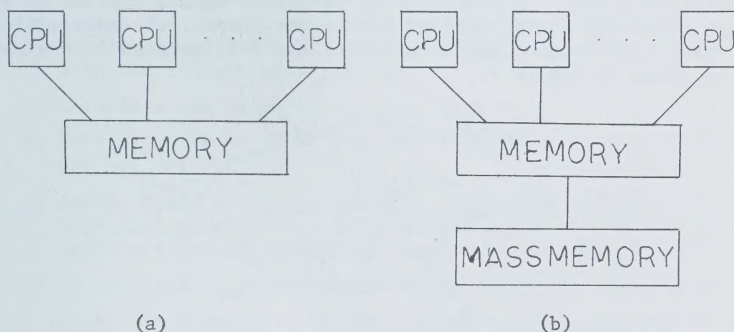


Figure 1. Models of Multiprocessor Systems

To summarize the formulation of the problem and the operation of multiprocessor systems, a computation graph and simple example is shown in Figure 2. The arcs represent tasks and are labeled with task execution times; precedence relations are indicated by the ordering of the graph. For example, task T_3 cannot begin until tasks T_1 and T_2 are completed. Precedence relations are written $T_1 \rightarrow T_3$; T_1 is called the ancestor of T_3 , and T_3 the descendent of T_1 . Immediate precedence relations are written $T_1 \rightarrow T_3$. A task list, L , based on an urgency strategy is derived as shown by Manacher (Ref. 4). This task list serves as a priority list in the operation of the multiprocessor system: as soon as a processor becomes

idle, the list is scanned from left to right, and the leftmost (highest priority) task all of whose ancestors have been completed is picked up and executed. The anomalies and stabilization problem associated with this approach have been adequately treated by Richards, Graham and Manacher.

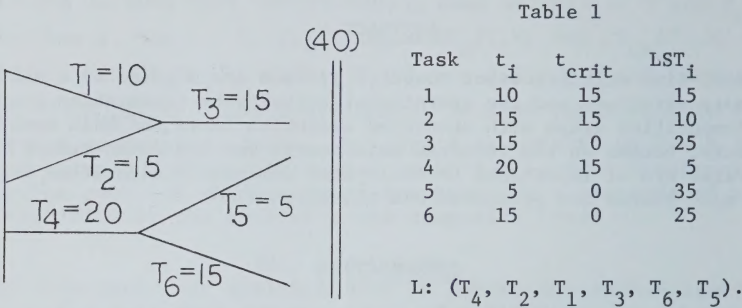


Figure 2. A Simple Example

Previous workers have assumed the existence of deadlines for individual tasks or the overall execution of a computation graph. Here, we shall assume that the shortest overall execution time is desired; our results also apply directly to the case of imposed deadlines. We thus begin by finding the shortest execution time by identifying the longest path in the computation graph. (A simple graph reduction method similar to signal flow graph reduction can be used or for larger graphs, the more efficient shortest path algorithms could be modified for this purpose.) A simple example is shown in Figure 3.

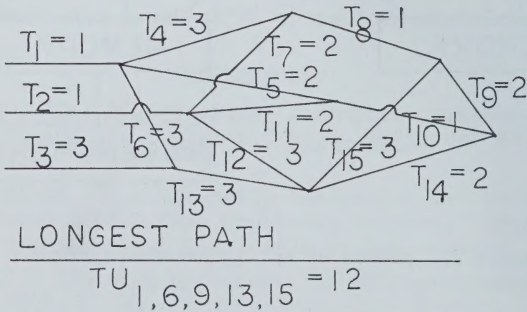


Figure 3. Example of Longest Path

Assuming there are no limitations on the number of processors, the memory size or the channel speed, the existence of a schedule yielding an execution time equal to the longest path can be shown in a straightforward manner. Successive extractions of longest paths can be performed, and each path can be assigned to a different processor. However, assuming the

minimum execution time is to be realized, the derivation of a schedule using a minimum number of processors is more interesting.

MINIMUM NUMBER OF PROCESSORS ASSIGNMENT

First, note that for a fixed overall execution period, T , the smallest number of processors, n_0 , is clearly bounded by:

$$n_0 \geq \lceil \Sigma t_i / T \rceil \quad (1)$$

where Σt_i is the sum of all task execution times and $\lceil x \rceil$ is the smallest integer $\geq x$. n_0 is the starting point for the generation of an assignment using the smallest number of processors.

Def. 1. The earliest starting time (EST) of a task T_i is the time, as defined by deletion of the longest path, prior to which T_i cannot be initiated. Similarly, the lastest starting time (LST) is defined. EST_i and LST_i define an interval to which the initiation of a certain task is restricted, called the initiation interval.

Def. 2. Let $T_{\leftarrow j}$ be the set of tasks to the left of the j -th task in the task list and $T_{\rightarrow j}$ the set of tasks to the right.

Def. 3. Let T_{-j} be the set of tasks picked up before T_j during execution and T_{+j} the set of tasks picked up after T_j .

Algorithm for Scheduling Tasks onto Minimum Number of Processors

- 1) Assign the longest path onto one processor, processor 1. Impose deadlines and earliest starting times on the computation graph with the longest path removed.
- 2) Calculate the value of n_0 , and let $n = n_0$.
- 3) Form the urgency table with the starting time constraints imposed, and the urgency task list.
- 4) Assign tasks according to their initiation intervals from the task list to $n-1$ processors following the method indicated in the Introduction. End when all tasks are assigned.
- 5) If deadline is exceeded when T_j is assigned, check whether $T_{\rightarrow j} \cap T_{-j} = \emptyset$ (null). If this relation holds, proceed to (6); otherwise, delete the lowest priority task which is in $T_{\rightarrow j}$ and in T_{-j} , and try a lower urgency task. If no lower urgency tasks can be initiated, leave the space idle. Branch to (4).
- 6) Replace n with $n + 1$; repeat process starting from (4).

Example 1. The computation graph in Figure 3 is repeated here. After extraction of the longest path, Figure 4(a) shows the remaining computation graph, and the EST's and LST's. Table 2 is the corresponding urgency table.

The task list formed from the urgency table is shown as:

L: (3, 2, 12, 4, 7, 5, 8, 11, 14, 10).

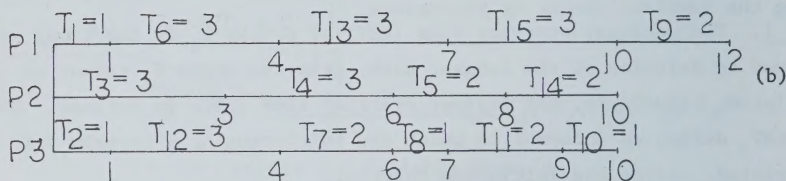
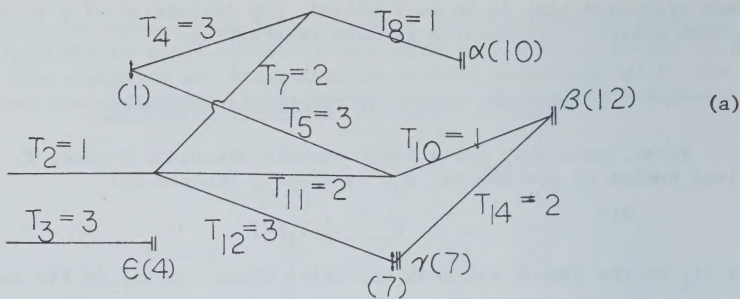


Figure 4. A Schedule for the Computation Graph Shown in Figure 3

Table 2. The Urgency Table of Example 1

EST_i	Task	$t_{crit\alpha}$	$t_{crit\beta}$	$t_{crit\gamma}$	$t_{crit\epsilon}$	$LST_{i\alpha}$	$LST_{i\beta}$	$LST_{i\gamma}$	LST
0	2	3	3	3		6	8	3	3
0	3				0				1
1	4	1				6			6
1	5		1				9		9
1	7	1				7			7
4	8	0				9			9
4	10		0				11		11
1	11		1				9		9
1	12		2	0			7	4	4
7	14		0				10		10

The sum of all the task time periods is given as

$$s = \sum_{i=1}^{15} t_i = 32 ; T = 12$$

$$n_o = [32/12] = 3$$

The assignment is carried out as shown in Figure 4(b). The PL (Ref. 4) is:

PL: (1, 3, 2, 6, 12, 4, 13, 7, 5, 8, 15, 11, 14, 10, 9)

END

A theorem follows directly from the above algorithm.

Theorem 1. If $T_{-i}^{\Delta T} = \phi$ and the latest starting time of T_i is

exceeded, then at least one more processor is required.

MINIMAL MEMORY AND CHANNEL SPEED REQUIREMENTS

We now introduce a second parameter in the description of a task: the size of the main core memory, m_j , required to execute the task T_j . It is convenient to describe the system operation in terms of a segmented memory; thus, the memory requirement will be specified in terms of pages, and the channel speed in terms of pages per unit time.

At any specific time, it is clear that the total main memory requirement is at least the summation of all the m_i 's of the tasks being executed. During transition points, i.e., those instants when at least one task is completed and another initiated, the instantaneous memory requirement must include the memory requirements of both the terminating task(s) and the initiating task(s). (We are of course assuming the time for switchover between tasks is negligible compared to execution time requirements.) A convenient format for indicating the fluctuating memory requirements is a memory occupation graph: let the vertical axis represent memory size and the horizontal, time; for a given schedule, a broken line curve represents the amount of memory used at any time, with asterisks representing the memory requirements at the transition or switchover points. (See Figure 5.) Clearly, the highest asterisk represents the minimum memory size, M_o .

Def. 4. For a given schedule and fixed main memory size of M_o , the minimum value of the channel speed which will guarantee nondelayed execution is called the feasible channel speed, C_f .

Turning now to the analogous case of minimum channel speed, C_o , we first assume that all of the starting tasks, that is, those tasks executed first by the system, are loaded into main memory beforehand. This initial system loading time could of course be easily determined once the channel speed is fixed. The task list with these starting tasks removed is called the modified task list. Consider a task T_i in the modified PL task list; at the time of initiation of this task, s_i , all of the tasks before T_i in PL must have been transmitted from secondary memory. The channel speed necessary to satisfy this requirement is clearly:

$$C_i = \sum_{j=1}^i m_j / s_i \quad (2)$$

where the i 's are indexed according to the modified PL list. The minimum channel speed, C_o is thus the maximum of the C_i :

$$C_o = \text{Max} \{C_1, \dots, C_m\} \quad (3)$$

where m is the largest index in the modified PL. The corresponding memory size which guarantees nondelayed execution for given PL and C_o is the feasible memory size, M_f .

We therefore turn to the problem of determining C_f and M_f . On the memory occupation graph, a horizontal line represents the minimum memory size, and an inclined line indicates, by its slope, the channel speed. A vertical line at a transition point indicates the amount of memory released. The combination of inclined and vertical lines represent the variation in the amount actively used memory. Together with the fixed memory line and the asterisks, we therefore have a vivid picture of the interaction between

task execution and memory occupation. C_f and M_f can then be determined by simple graphical constructions.

Assume the channel transmits the tasks one by one into the main memory following the modified PL list. Whenever the fixed memory size boundary is reached, the channel becomes idle until an amount of memory is released upon completion of a task by a processor. The relative position of the asterisks to the memory occupation curve indicates whether or not the channel speed or the memory size is sufficiently large: the curve indicates the amount of information which can be transmitted while the asterisks indicate the amount of information required to accommodate the instantaneous task load.

Algorithm for Determination of Feasible Channel Speed

- (1) Let the minimum channel speed, C_0 be the initial trial value, and the minimum memory, M_0 be the boundary.
- (2) Plot the inclined line with slope equal to the trial value. If the memory boundary is reached, let the channel idle until some memory is released. Take the releasing of the memory into account by introducing a drop along the curve. During the process, each asterisk must be beneath the curve. If not, proceed to (3); otherwise, remain in (2) until the last asterisk is passed. The last trial value is the feasible channel speed, C_f .
- (3) Let M_* be memory required by asterisk point above trial memory curve and M' be memory of point on curve under asterisk. Calculate the quotient of $M_* - M'$ divided by the time elapsed from the last channel idling period, and add quotient to the trial value to form a new trial value. Branch to (2).

The counterpart of the above method is to keep the channel speed constant and to vary the memory size.

Algorithm of Determination of Feasible Memory Size

- (1) Let M_0 be the initial trial value.
- (2) Use value C_0 to move tasks into the core. As long as the curve is above the asterisks, no change of memory size is required; otherwise, branch to (3). If all the asterisks are under the curve, the last trial value is the feasible memory size.
- (3) Increase the trial value by the difference $M_* - M'$. Branch to (2).

Example 2. The schedule generated in Example 1 is used to find the two minimum values and the two feasible values.

From Figure 5(b), we obtain $M_0 = 32$ pages.

Table 3 shows the generation of the required channel speed:

$$C_0 = 10 \text{ pages/unit time}$$

At $t = 7$

difference: $32 - 30 = 2$ pages

time elapsed: $7 - 6 = 1$ time unit

new trial value: $10 + (2/1) = 12$ pages/unit time

$C_f = 12$ pages/time unit

On the other hand,

$M_f = 34$ pages

END

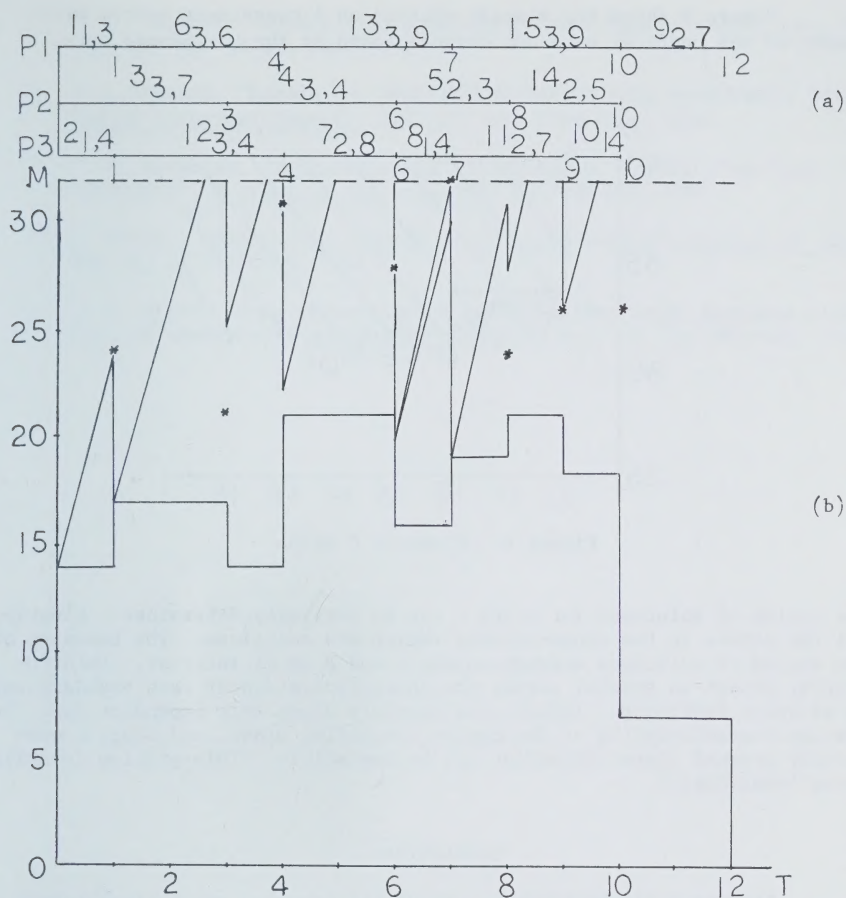


Figure 5. The Memory Occupation Graph

Table 3. Generation of the Minimum Channel Speed

Index	Task	Summation	Initiation Time	Required C
1	6	6	1	6
2	12	10	1	10
3	4	14	3	4.66
4	13	23	4	5.75
5	7	31	4	7.75
6	5	34	6	5.66
7	8	38	6	6.33
8	15	47	7	6.71
9	11	54	7	7.71
10	14	59	8	7.375
11	10	63	9	7
12	9	70	10	7

Figure 6 shows the results plotted on a coordinate system with memory as the vertical axis and channel speed as the horizontal axis.

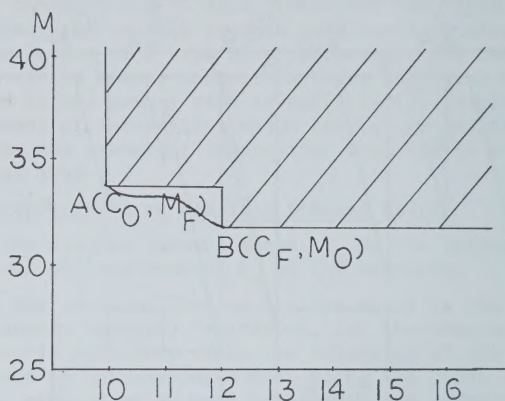


Figure 6. M versus C Graph

The region of solutions for M and C can be partially determined. Clearly, all the points in the cross-hatched region are solutions. The boundary of the region of solutions between points A and B is of interest. Unfortunately, except in special cases, the characterization of this boundary seems to be quite difficult. Indeed, the boundary seems very dependent upon the precise characteristics of the memory occupation graph, and thus, a completely general characterization may be impossible. This problem is still being investigated.

CONCLUSIONS

Through a slight modification of the hardware model of multiprocessors, we have been able to extend previous scheduling methods to allow determination of bounds on memory and channel speed requirements. Two types of bounds were found: one with minimum memory and sufficiently large channel speed to accommodate this memory, and the analogous case of minimum channel speed and corresponding memory size. Clearly, a family of solutions exists to this problem. The extension of the analysis to allow determination of an optimal solution, say with respect to some reasonable cost assumptions, is an interesting open problem. We might note that this problem is primarily of analytical interest, since our results allow determination of solutions satisfying the usual practical constraints on memory size and/or channel speed. That is, if one of the system parameters M or C is fixed, the minimal value of the other for that specific constrained value is obtainable.

REFERENCES

1. P. Richards, "Timing Properties of Multiprocessing Systems", Tech. Paper, Rep. No. TD-B60-27, Tech. Operations, Inc., Burlington, Mass., Aug. 1964.

2. B. P. Ochsner, "Controlling A Multiprocessor System", Bell Labs. Rec., Vol. 44, pp. 59-62, Feb. 1966.
3. R. L. Graham, "Bounds for Certain Multiprocessing Anomalies", Bell System Technical Journal, Vol. 45, pp. 1563-1581, 1966.
4. G. K. Manacher, "Production and Stabilization of Real-Time Task Schedules", Journal of ACM, Vol. 14, pp. 439-465, 1967.
5. J. Heller, "Sequencing Aspects of Multiprocessing", Journal of ACM, Vol. 8, pp. 426-439, 1961.
6. J. D. Wiest, "Some Properties of Schedules for Large Projects with Limited Resources", Operations Research, Vol. 12, pp. 395-418, 1964

CONTROLLABILITY OF NONLINEAR SEQUENTIAL NETWORKS

FREDERIC J. MOWLE
Purdue University
Lafayette, Indiana

ABSTRACT

A sequential network is said to be controllable if there exists at least one integer k such that it is possible to transition between any pair of arbitrary states (S_i, S_f) in exactly k steps. In this paper, necessary and sufficient conditions are given for a nonlinear sequential network to be controllable.

INTRODUCTION

Martin Cohn, in his paper "Controllability on Linear Sequential Networks" [1] defined a Sequential Network to be controllable if there exists at least one integer k such that it is possible to transition between any pair of arbitrary states (S_i, S_f) in exactly k steps. Cohn showed that for the linear case, strong connectedness was both necessary and sufficient for controllability.

For the nonlinear case, strong connectedness is necessary, but no longer sufficient as is easily seen from Figure 1.

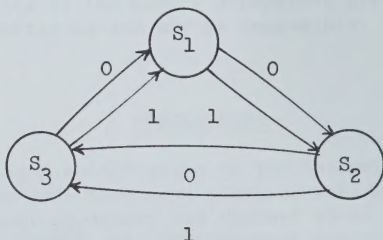


Figure 1: State diagram of a nonlinear sequential machine that is not controllable.

We can number the states on the sequential machine in some arbitrary manner denoting the j th state by S_j . We can then use the following vector notation:

$$\underline{g} = (g_0, g_1, g_2, \dots, g_{n-1})$$

to denote a set of states from the machine where

$$\begin{aligned} g_i &= 1 \text{ if state } S_i \text{ is } \in G \\ &= 0 \text{ if state } S_i \text{ is } \notin G \end{aligned}$$

We will let θ be the next state operator [2]. Thus if $\underline{g}$ is the present set

of a state under consideration, θG will denote the set of all possible successors of the states in G under all inputs, and $\theta^j G$ will denote the j^{th} set of possible successors of the states in G under all inputs.

Let us denote the single state set $\{i\}$ consisting of state S_i by

$$\underline{G}_i = (0, \dots, 0, 1, 0, \dots, 0),$$

all zeros except in position i .

Let us also denote the set of all states by the vector $\underline{1}$, the vector whose components are all ones.

CONTROLLABILITY

Theorem 1. A sequential network which is strongly connected and contains at least one equilibrium state is controllable.[1]

Proof. Starting in any state S^* there exists a path to an equilibrium state, and a path from the same equilibrium state to S^{**} . The step difference required to assure reaching S^{**} in exactly k transitions from all states can be obtained by delaying at the equilibrium state. Q.E.D.

It is clear that Theorem 1 yields only a sufficient condition for controllability. It is obvious that a necessary but not sufficient condition for a sequential network to be controllable is that it be strongly connected.

A necessary and sufficient condition for controllability can be stated as follows:

Theorem 2. If there exists an integer k_i for each state S_i such that $\theta^{k_i} \underline{G}_i = \underline{1}$ for all $i=0, \dots, n-1$, where $\underline{G}_i$ is the set vector for $\{i\}$, then the network is controllable.

Proof. Necessity: Assume for some state i there is no such integer k_i . Then by definition the network is not controllable.

Sufficiency: Let us separate out the initial state, S^* , and the terminal state, S^{**} , from the remaining states, and show that the network is k controllable with

$$k = \sum_{i=0}^{n-1} k_i.$$

(refer to Appendix A for the remainder of the proof)

Corollary 1. The minimum k for controllability is equal the $\max(k_i)$, $i=0, \dots, n-1$.

Proof. Since the network is controllable, k_i 's exist for each state. Also, since the machine is controllable, it is strongly connected; thus $\theta \underline{1} = \underline{1}$. Hence, if $k_\alpha < k$, there exists an input sequence of length $k - k_\alpha$ starting from at least one state, say S_β , which leads to the terminal state S^{**} or else we are contradicting the fact that the network is strongly connected.

We know an input sequence of length k_α exists which will take us from S^{**} to S_β ; thus there exists input sequences of length $k = \max(k_i)$, $i=0, \dots, n-1$ which will allow transitions between any pair of states S^* and S^{**} .

Corollary 2. The sequential network which is k controllable with the minimum $k = \max(k_i)$ is controllable for all $k \geq \max(k_i)$.

Proof follows from the proof of corollary 1.

The two theorems and corollaries above lead to the following brute force procedure for testing a sequential network to see if it is controllable.

TEST PROCEDURE - BRUTE FORCE

Test for Strong Connectedness. If $\theta \underline{1} \neq \underline{1}$, the sequential network is not strongly connected. If $\theta \underline{1} = \underline{1}$, set $\{G\} = \{i\}$, $i=0, \dots, n$, $\underline{G} = \underline{I} = (0, \dots, 0, 1, 0, \dots, 0)$ (all zeroes except for the i^{th} position). For each set $\{i\}$ calculate $\hat{\theta}^j \underline{I}$, $j=1, \dots$ stopping when $\hat{\theta}^{j+1} \underline{I} = \hat{\theta}^j \underline{I}$. If $\hat{\theta}^j \underline{I}$ is the all one vector for each $\underline{I}$, then the network is strongly connected. Note $\hat{\theta}^j \underline{I} = \bigvee_{i=0}^j \theta^i \underline{I}$; i.e., $\hat{\theta}^j \underline{I}$ is the logical OR of the j^{th} next state sets with the original set $\{I\}$.

Test for Controllability.

- (a) Check to be sure the network is strongly connected.
- (b) If the network is strongly connected and contains at least one equilibrium state, then the NSN is controllable.
- (c) If the network is strongly connected but contains no equilibrium state, then the following brute force method can be used.

Set $\{G\} = \{i\}$ $i=0, \dots, n-1$

$\underline{G} = \underline{I} = (0, \dots, 0, 1, 0, \dots, 0)$ (the all zero vector except for the i^{th} position)

For each set $\{i\}$ calculate $\theta^j \underline{I}$, $j=1, \dots$ stopping when one of the following occur:

- (1) $\theta^{j+1} \underline{I} = \theta^j \underline{I}$
- (2) $\theta^j \underline{I} = \underline{1}$ (all one vector)
- (3) $j = 2^n$

If $\theta^j \underline{I} = \underline{1}$ (all one vector) for each state set $\underline{I}$, then the machine is controllable.

The following nonlinear sequential network is controllable for all $k \geq 4$. The controllability calculations may be found in Table 1.

Example.

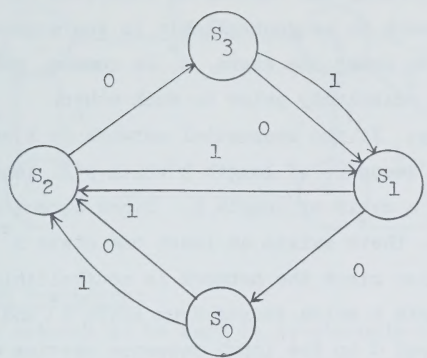


Figure 2: State diagram of a controllable sequential network.

Table 1

i	=	0	1	2	3
$\underline{G}_i$		[1, 0, 0, 0]	[0, 1, 0, 0]	[0, 0, 1, 0]	[0, 0, 0, 1]
$\theta \underline{G}_i$		[0, 0, 1, 0]	[1, 0, 1, 0]	[0, 1, 0, 1]	[0, 1, 0, 0]
$\theta^2 \underline{G}_i$		[0, 1, 0, 1]	[0, 1, 1, 1]	[1, 1, 1, 0]	[1, 0, 1, 0]
$\theta^3 \underline{G}_i$		[1, 1, 1, 0]	[1, 1, 1, 1]	[1, 1, 1, 1]	[0, 1, 1, 1]
$\theta^4 \underline{G}_i$		[1, 1, 1, 1]			[1, 1, 1, 1]
		$k_0 = 4$	$k_1 = 3$	$k_2 = 3$	$k_3 = 4$

If we denote the input sequence by x_1, x_2, x_3, x_4 then one possible set of inputs for this network is shown in Table 2.

Table 2
Terminal State

initial state	0	1	2	3
0	[0, 0, 0, 0]	[0, 1, 1, 1]	[0, 0, 0, 1]	[0, 1, 1, 0]
1	[0, 0, 1, 0]	[0, 0, 0, 0]	[0, 0, 1, 1]	[1, 1, 1, 0]
2	[1, 1, 1, 0]	[0, 0, 1, 1]	[0, 0, 0, 0]	[1, 0, 0, 0]
3	[0, 1, 1, 0]	[0, 1, 0, 0]	[0, 1, 1, 1]	[0, 0, 0, 0]

HEURISTIC APPROACH

The previous theorems have lead to a time-consuming brute force test procedure. Let us now consider ways to decrease the testing time.

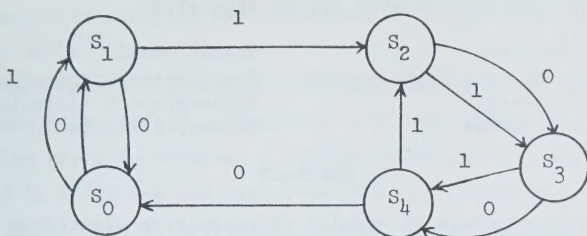
Theorem 3. A necessary and sufficient condition for a strongly connected sequential network to be controllable is the existence of two cycles C_1 and C_2 which have at least one state, S^* in common, provided the cycle lengths L_1 and L_2 are relatively prime to each other.

Proof. Necessity: If the sequential network is k controllable, then there exists an input sequence of length k which will lead from some state S^* back to S^* producing a cycle of length k . Since a controllable network is strongly connected, there exists at least one state S^{**} such that $\theta S^{**} = S^*$ for some input α . Also since the network is controllable, there exists an input sequence of length k which starts from state S^* and terminates on state S^{**} . Adding input α to the input sequence carries us back to state S^* and yields a new cycle of length $k+1$. Thus we have shown the existence of a state S^* common to two cycles whose cycle lengths are relatively prime.

Sufficiency: A lemma [3] from number theory states that one property of the greatest common divisor is given any two positive integers $K_1 > 0$ and $K_2 > 0$, there exists two other integers N_1 and N_2 (not necessarily positive) such that the greatest common divisor D can be expressed as $D \equiv K_1 N_1 + K_2 N_2$. In the case of relatively prime numbers $D = 1$.

Using this lemma, we can always form two cycles C_1^* and C_2^* whose cycle lengths L_1^* and L_2^* are multiples of the original cycles and such that $|L_1^* - L_2^*| = 1$. Therefore, without loss of generality, we need only consider the case $|L_1 - L_2| = 1$. Let us also relabel the longer cycle C_g and its length L_g and the shorter cycle C_s and its length L_s . Since the network is strongly connected, there exists a path from every state to S^* , the state in common, and also there exists a path from S^* to every state. Hence, there exists a path of minimum length between every pair of states subject to the constraint that the path includes the state S^* . Let us label the shortest such path between some pair of states $L_{\min}$ and the longest $L_{\max}$. Let us also label the length of the shortest path between the pair of states S_i and S_j , L_{ij} . When we now transition between states S_i and S_j by way of S^* we will stop at state S^* and append the cycle C_g a total of $(L_{\max} - L_{ij})$ times and the cycle C_s a total of $(L_{ij} - L_{\min})$ times. Using this procedure, the total path length between any pair of states will be equal to $L_{\min} + L_g(L_{\max} - L_{\min})$. Thus the network is indeed controllable.

Example: The sequential network shown in Figure 2 can be checked for controllability by inspection using Theorem 3. In the following diagram, states S_0 , S_1 , S_2 and S_4 form a cycle of length 4 and states S_2 , S_3 and S_4 form a cycle of length 3. The two cycle lengths are relatively prime and state S_2 is common between them.



Heuristic Test Procedure:

- (a) Test the network to be sure it is strongly connected.
- (b) Search the network for two cycles with at least one state in common whose cycle lengths are relatively prime.

CONCLUSIONS

In this paper both brute force and heuristic procedures for testing a sequential network for controllability have been given. At present, the latter approach appears to be the most useful and should help determine if complete classes of sequential networks are controllable. Additional work remains to be done before we can quickly determine the necessary input sequences for a nonlinear sequential network.

REFERENCES

1. M. Cohn, "Controllability in Linear Sequential Networks," IRE Transactions on Circuit Theory, Vol. CT-9, No. 1, pp. 74, 78, March 1962.
2. F. J. Mowle, "An Analysis Procedure for Feedback Shift Registers," Eleventh Midwest Symposium on Circuit Theory, University of Notre Dame, pp. 601-610, May 1968.
3. Birkhoff and MacLane, A Survey of Modern Algebra, 3rd Edition, MacMillan, New York, 1965.

APPENDIX A

Let us start in the initial state S_1^* and proceed to any unused state S_j except the terminal state S^{**} using an input sequence of length k_1 . From this new starting state, say S_α , we can proceed to any unused state except S^{**} using an input sequence of length k_α . We can continue in this way until we arrive at some state, say S_β , with only terminal state S^{**} unused. By applying an input sequence of length k_β we can reach the terminal state. The total input sequence is of length

$$k = \sum_{i=0}^{n-1} k_i.$$

MODULE COMPLEXITY AND NAND NETWORK DESIGN ALGORITHMS

EDWARD S. DAVIDSON
Stanford Electronics Laboratories
Stanford University
Stanford, California

GERNOT METZE
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois

ABSTRACT

The effectiveness of modular decomposition algorithms for combinational switching functions is shown by argument and by example to decrease with increased module complexity. Module selection for algorithms is discussed for circuit environments of individual NAND gates, small arrays, and large arrays of NAND gates.

INTRODUCTION

Logic design in an all NAND gate circuit environment was prevalent early in the history of computer design. Now with the advent of integrated circuits, the trend back to NAND design is already well-established. The NAND function is as basic to logic design, by virtue of its completeness, as it is to circuit fabrication, by virtue of its direct relation to the simplest of circuits. Also, the possibility of presenting a regular array of identical gates, plus flip-flops, on a chip is economically attractive. It is not unlikely that more complex "gates" used by the logic designer would be built up from NAND gates by the circuit designer at the most basic level. The principal disadvantage of working only with NAND gates lies in the apparently non-intuitive nature of the design of NAND networks. Programmable algorithmic procedures for NAND network design are thus essential, particularly for design automation systems in an LSI circuit environment. For existing algorithms, contrasting decisions with respect to module selection have been made. It is the purpose of this paper to discuss these decisions from the point of view of applying the resulting algorithms to an individual NAND gate environment and to an environment of small standard interconnected arrays of NAND gates. The conclusions drawn are then extended to an environment of large arrays of NAND gates with non-standard interconnections.

EXISTING ALGORITHMS

Algorithmic combinational network design begins with a set of functions to be realized as output functions of a network. The task of the algorithm is to decompose the given functions into sub-functions with each sub-function corresponding to an available building block. The form

of the decomposition dictates the interconnections between the building blocks in the solution network.

A NAND decomposition algorithm has been proposed by Davidson [1] [3] which deals directly and exclusively with NAND gates as its building blocks. At the opposite extreme, a modular decomposition algorithm has been proposed by Schneider and Dietmeyer [5] which deals with a library of multi-input, multi-output, internally interconnected modules as building blocks. Typically, 3-input, 2-output modules, equivalent to 2 to 4 NOR gates, are used. Between the two extremes, with respect to module selection, lies an algorithm developed by Karp, McFarlin, Roth, and Wilts [4] which uses a collection of "vertex functions," i.e. functions which attain the value 1 for either one input vector or all but one input vector, as modules. Typically, a subset of AND, OR, NAND, NOR, and NOT is used.

The progress of each of these algorithms may be represented by a solution tree. The root node of the tree represents all solutions to the problem. Branching from a node represents the alternatives available at an iteration of the algorithm. A decision is made at each iteration to explore one of the alternatives causing branching to occur next from the node of the selected alternative and adding a level to the paths of the tree containing that node. Backtracking in the solution tree may be used to reexamine previously unexamined alternatives.

The NAND decomposition approach is apparently well-matched to an actual circuit environment of individual NAND gates, i.e. an environment in which only NAND gates are available and in which the cost of a solution network is directly related to the number of gates and gate inputs used, although network constraints may be incorporated. Nevertheless, even in such an environment, a modular decomposition approach may be applied yielding a network of suitably chosen modules each of which would be replaced by an equivalent NAND network. Such an indirect approach would be justified if modular decomposition were substantially more efficient than NAND decomposition, so that a reduction in algorithm running time would be possible, and if fully optimum solutions were not required.

On the other hand, modular decomposition is apparently well-matched to an environment of small arrays of NAND gates with standard interconnections within the arrays. Each module in the library could correspond to a standard array. Yet, NAND decomposition could be applied to this environment, provided a packaging algorithm were available to

package the resulting NAND network into the small arrays. Such an approach would be justified if NAND decomposition were substantially faster than modular decomposition, and if packaging can be accomplished efficiently.

RELATIVE EFFICIENCY OF ALGORITHMS

Let us then examine the relative efficiency of NAND decomposition and modular decomposition. An iteration of the NAND decomposition algorithm roughly corresponds to assigning an input to a gate.[†] A crude bound on the number of assignment alternatives is $v + g$ where v is the number of variables and g the number of gates presently in the partial solution. The actual number of alternatives would be much less since only a few gates and variables are connectible as inputs and fewer yet may perform the covering task upon which the iteration is based. The algorithm then produces a solution tree with roughly as many levels as the number of interconnections of a solution network and with the number of branches from each node certainly no more than the number of variables and gates of the solution. The number of alternatives at each decision point is thus well in hand, so that the immediate consequences of each alternative may be carefully evaluated before a decision is made. The algorithm is, however, highly dependent upon extensive tree pruning in backtracking, since a full expansion of the solution tree according to the bounds presented is unthinkable.

The problem becomes somewhat different as the size and number of available modules grow, as in modular decomposition. If there are L modules in the library, each with P outputs and I inputs, then an iteration of a v variable problem with M modules in the partial solution presents $L(v + 2 + P \cdot M)^I$ alternatives. This bound results from selecting I inputs at a time from the v variables, 2 constants, and $P \cdot M$ previously selected module outputs. This process could be carried out for each of the L library modules. Thus the number of alternatives at an iteration could be quite large, e.g. for $L = 10$, $v = 6$, $P = 2$, $M = 3$, $I = 3$ the bound is 27,440. Of course, only a small fraction of these are meaningful. Many would be so degenerate as to be obviously useless; many can be rejected by a number of algebraic arguments based on their relationship to alternatives already tested. Nevertheless, a severe problem remains in selecting an

[†] Actually a particular iteration may not assign any new inputs or may assign several inputs and add new gates to the partial solution.

ordered collection of inputs and in selecting a module from the library. In addition, the number of meaningful alternatives is generally so large that it is impractical to evaluate all alternatives before selecting one. Thus a heuristic must be found which measures whether an evaluated alternative should be accepted or whether further evaluation of alternatives should be undertaken. Although the potential branching from a node of the solution tree is extremely large, the number of levels of the tree is relatively small, e.g. for i -input gates and g gates per module, the number of levels of a NAND decomposition tree is $i \cdot g$ times the number of levels of a modular decomposition tree. Even with the fewer levels, the increased number of alternatives at each node makes full expansion of the modular decomposition tree to the bounds given as unthinkable as with NAND decomposition.

NAND decomposition gains efficiency if many tree paths can be abandoned after being expanded for only a few levels. Modular decomposition gains efficiency if the alternative selection heuristics select good alternatives after considering very few of the alternatives and in addition, if very few of the alternatives need to be expanded in backtracking.

TWO EXAMPLES

The case for NAND decomposition being more efficient is supported by two examples.

The first example is taken from Schneider and Dietmeyer [5]. Four functions, y_1 to y_4 , of five variables, x_1 to x_5 , are to be realized, as shown in Fig. 1. The modules are shown in Fig. 2. In the Schneider and Dietmeyer algorithm, derived modules, such as α_{11} and α_{12} from α_{10} , are added to the library. Assigning inputs to constant functions and assigning two or more inputs to the same function are then forbidden. With this convention, some obviously useless input assignments are avoided. Other modules derived from α_{10} and modules derived from α_{20} and α_{40} are not shown. All gates are NOR gates except for the NAND gate in α_{20} . Cost is the number of gates in the modules used. Delay is computed by Schneider and Dietmeyer on a module basis, associating a delay for each module (S & D delay) equal to the maximum path, measured in gates, through the module. The actual delay of each output, measured in gates, is also shown.

The Schneider and Dietmeyer modular decomposition algorithm, programmed in FORTRAN for an IBM 1620 Model II computer, produced the solution of Fig. 3 in 2 hours and 18 minutes without backtracking. Thus,

while considerable numbers of alternatives must have been examined, only five decisions in the solution tree were made: one for each module used. In all probability, far fewer alternatives were examined, even for those five nodes, than would be necessary if backtracking were attempted. The solution of Fig. 3 had minimum S & D delay as its design goal. The solution of Fig. 4 was designed by their modular decomposition to minimize cost. Due to the simple relation of the modules used to NOR gate equivalents, the dual problem was presented to the NAND decomposition algorithm to achieve an all NOR gate solution. The minimum gate solution found is shown in Fig. 5.

The solution tree contained paths of up to 18 levels. The NAND decomposition algorithm was programmed in ILLAR, a macroassembly language, for a CDC 1604A computer. The solution of Fig. 5 was produced, with backtracking to prove optimality, in $4\frac{1}{2}$ seconds. While a precise conversion factor for comparing run times is not possible, it is clear that such a factor is substantially below the factor of 1800 achieved here. This factor would be raised still further if backtracking were attempted by Schneider and Dietmeyer to find an optimum solution and prove its optimality.

It might be argued that modular decomposition is not basically inefficient, but that the modules were merely ill-suited to this problem. This is in fact not the case as the modular packaging of the NOR gate network demonstrates, see Fig. 6. Even when the 8 gate solution is packaged, the cost is only 10 rather than the 14 cost of Fig. 4, the "minimum cost" modular decomposition solution. Furthermore, although the S & D delay of Fig. 6 is greater than the S & D delay of Fig. 3, the "minimum delay" modular decomposition solution, the actual delay of Fig. 6 is less than the actual delay of Fig. 3. The solution of Fig. 6 thus demonstrates that the modules are indeed better suited to the problem than demonstrated by Figs. 3 and 4. Second it might be argued that so many alternatives are evaluated by Schneider and Dietmeyer that backtracking is unnecessary, but a comparison of the costs of Fig. 6 and Fig. 4 disproves that argument. Thus in this example, NAND decomposition is by far the more efficient process.

The second example is taken from the paper by Karp, et al. [4]. Their use of only vertex function modules much reduces the number of alternatives at a decision node. Module inputs fall into but two symmetry classes, only one constant input is relevant to each module input, and further algebraic elimination of alternatives is done. However, due to

the simpler modules, more levels are required in the solution tree. Thus, this approach presents both the level control problem of NAND decomposition and the alternatives control problem of modular decomposition, but neither problem is as severe taken alone.

The algorithm was programmed for an IBM 7090. The 3-variable majority function was realized with NOR and $\cdot\text{AND}^\dagger$ building blocks in 9 seconds and possesses an all-NOR optimum solution. The same (self-dual) network with NAND gates was found to be optimum by NAND decomposition in 1 second. The longest realization and proof of optimality time required by Karp, et al. for any 3-variable function was 10 minutes. The comparable time for NAND decomposition is $\frac{1}{2}$ minute. Thus the NAND decomposition extreme appears more efficient than even a modular decomposition approach which uses only NOR and AND as modules and is thus very close to NAND decomposition.

NAND decomposition profits by working with only a single module. Often the number of alternatives at a potential decision node is 1: only addition of a new gate, in this case necessarily a NAND gate, will suffice to perform the covering task of the node. Such a node is collapsed into its predecessor node during alternative evaluation. The levels of the tree and the number of decision nodes are thus often reduced. Working with even two module types destroys this advantage.

CONCLUSIONS

In a circuit environment of individual NAND gates, the application of NAND decomposition is preferable to the application of modular decomposition, since NAND decomposition is inherently more efficient and is directly applicable to the environment

Small arrays of NAND gates connote here a collection of few enough NAND gates so that standard arrays, i.e. arrays with standard interconnection patterns within them, can be found. This environment is equivalent to any circuit environment of standard combinational logic packages, since any such package may be replaced by a small interconnected array of NAND gates. Modular decomposition is directly applicable to such an environment. NAND decomposition is only indirectly applicable with the use of a packaging algorithm. Furthermore, the sequence of NAND decomposition and packaging may yield badly non-optimal results. Yet NAND

[†] The $\cdot\text{AND}$ is a 0-cost AND function formed by connecting 2 or more NOR gate outputs (collectors) together.

decomposition is more efficient than modular decomposition. Which approach is preferable for this environment then depends on the efficiency of a packaging algorithm and its applicability to the set of standard arrays.

The applicability of a packaging algorithm would depend critically upon the available standard arrays. For example, if no interconnections were present within the arrays, an efficient packaging algorithm could be constructed. If in addition, the cost of a packaged solution were directly related to the number of gates and gate inputs of the solution, e.g. a piecewise linear combination of gates and gate inputs with non-negative coefficients, the application of NAND decomposition followed by packaging should be both efficient and near optimal. If it is necessary to include some interconnections within the arrays, sufficient generality of types of arrays should be provided so that the unpackaged network need not be highly non-optimum with respect to the specified cost function and under the specified network constraints in order to obtain an optimum packaged solution. Such would not be the case, for example if the only gate available with 4 or more inputs had no inputs accessible to the terminals of its array. However, if many 4 input gates were present in the set of standard arrays, all of them need not be free of internal interconnections. In view of the computational advantage of NAND decomposition over modular decomposition, it is suggested that modules might be selected to afford easy, near optimal packaging of NAND network solutions.

Large arrays are considered to be those of such size that sets of standard packages are unlikely. However, even if standard packages of several hundred gates were available, the computational difficulties of modular decomposition would make that approach prohibitive. Clearly a direct application of NAND decomposition to the logic of several large arrays is also infeasible. Thus the problem of NAND network design using non-standardized large arrays of gates involves a new problem: that of partitioning logic, which is not yet realized by a network, into array-size pieces. Within the array, since no interconnections are yet present, an individual NAND gate environment exists. If the number of gates of a single array is still too large to be treated efficiently by NAND decomposition, the partitioning algorithm used above could be applied within the array to produce sub-array size sets of logic functions which can be designed successfully by NAND decomposition.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant NSF GK-1663.

REFERENCES

1. Davidson, E. S., "An Algorithm for NAND Decomposition of Combinational Switching Functions," Report No. R-382, Coordinated Science Laboratory, University of Illinois, Urbana, June, 1968.
2. Davidson, E. S. and Metze, G., "Comments on 'An Algorithm for Synthesis of Multiple-Output Combinational Logic'," IEEE Trans. on Computers, to be published.
3. Davidson, E. S., "An Algorithm for NAND Decomposition Under Network Constraints," to be published.
4. Karp, R. M., McFarlin, F. E., Roth, J. P., and Wilts, J. R., "A Computer Program for the Synthesis of Combinational Switching Circuits," Proc. AIEE Symposium on Switching Circuit Theory and Logical Design, pp. 182-194, October 17-20, 1961.
5. Schneider, P. R. and Dietmeyer, D. L., "An Algorithm for Synthesis of Multiple-Output Combinational Logic," IEEE Trans. on Electronic Computers, Vol. C-17, No. 2, pp. 117-128, February, 1968.

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4
-	-	1	0	0	0	1	0	1
1	-	1	0	1	1	0	0	1
-	1	1	0	1	1	0	0	1
0	0	1	0	1	1	0	1	1
1	-	0	0	1	0	0	1	0
-	1	0	0	1	0	0	1	0
0	0	0	-	1	0	0	1	0
0	0	-	1	1	0	0	1	0
-	-	0	-	0	0	0	0	0
-	-	-	1	0	0	0	0	0
1	-	-	1	1	0	0	0	0
-	1	-	1	1	0	0	0	0

Fig. 1. Function to be synthesized

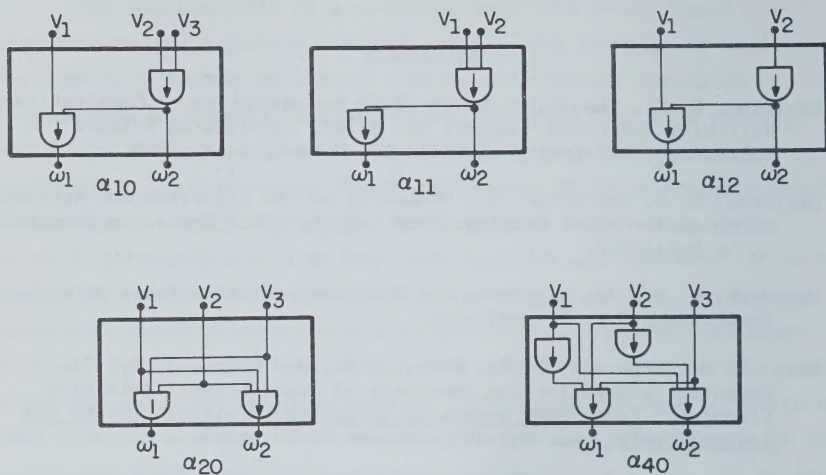


Fig. 2. Pertinent Modules from the Library of Functional Packages

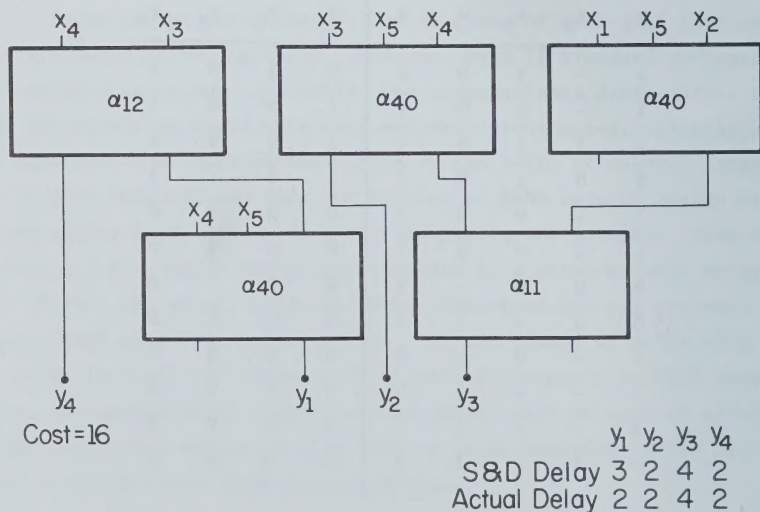


Fig. 3. A "Minimum Delay" Implementation

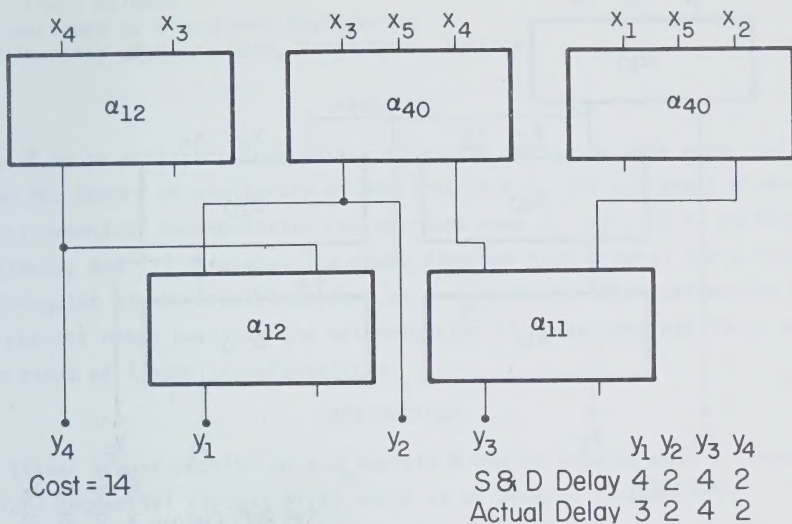
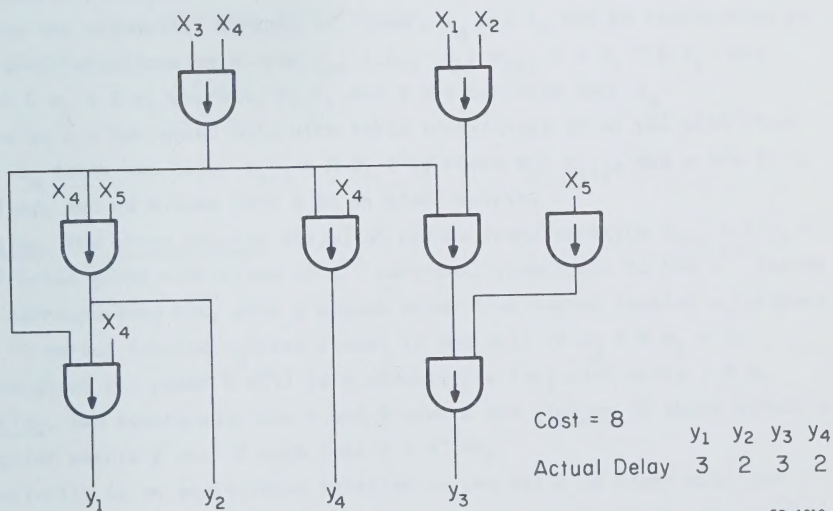


Fig. 4. A "Minimum Cost" Implementation



FR-1810

Fig. 5. Optimal Used Gate Implementation

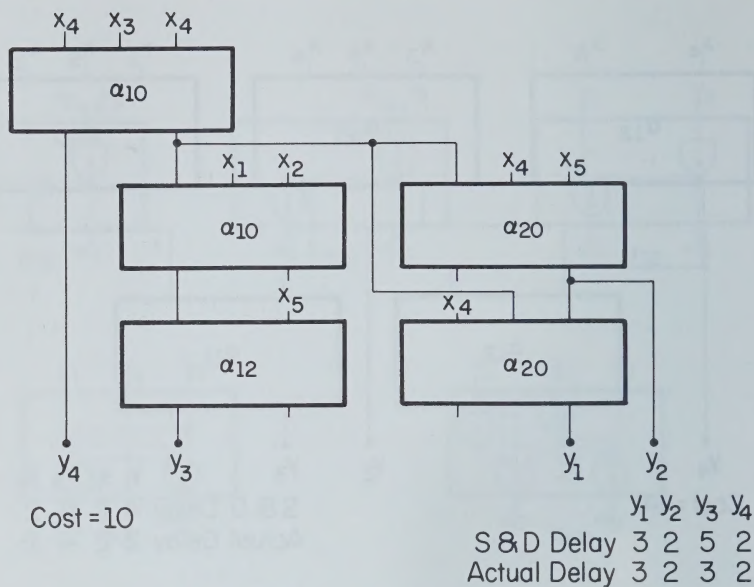


Fig. 6. Minimum Used Gate Implementation

STATE DIAGRAMS OF NONSINGULAR TRANSFORMATIONS ON J_m -MODULES

H. LOUIS ALTHAUS

Department of Electrical Engineering
University of Notre Dame, Notre Dame, Indiana

ABSTRACT

Let R be an arbitrary commutative ring with unit. In this paper we (1) develop the theory of similarity of matrices over R , (2) outline a primary rational canonical decomposition for matrices over J_m (subject to certain conditions), and (3) determine the state diagrams (cycle sets) for a class of nonsingular linear transformations on a J_m -module. The combination of these results makes possible the determination of state diagrams for a much larger class of linear transformations.

INTRODUCTION

A linear transformation on a J_m -module M can be equated with a linear (modular) sequential circuit [1,2] which is an ordered 5-tuple:

$$LSC = \{ M, X, Y, \delta_M, \lambda_M \}$$

where M , X , and Y are finite dimensional modules over J_m such that M is a set of states ($\sigma \in M$), X is a set of inputs ($x \in X$), and Y is a set of outputs ($y \in Y$). $\delta_M: M \times X \rightarrow M$ is the transition function and $\lambda_M: M \times X \rightarrow Y$ is the output function.

Since the sequential circuit is linear, δ_M and λ_M can be represented as matrix transformations on M over J_m , i.e., $\delta_M: \sigma_{i+1} = A \sigma_i + B x_i$ and $\lambda_M: y_i = C \sigma_i + D x_i$ where A , B , C , and D are matrices over J_m .

Here we are concerned only with state transitions of an LSC with fixed input. δ_M takes the form: $\sigma_{i+1} = A \sigma_i + \omega$, where σ_i , σ_{i+1} , and ω are in M , ω is fixed, and we assume that A is an $h \times h$ matrix.

Definition. The state diagram $G(A, \omega)$ of affine transformation $\sigma_{i+1} = A \sigma_i + \omega$ is a directed graph consisting of $m^{h'}$ vertices, isomorphic to the $m^{h'}$ states of the corresponding LSC, with a branch drawn from vertex labeled σ_i (present state) to vertex labeled σ_j (next state) if and only if $\sigma_j = A \sigma_i + \omega$.

Throughout the paper R will be a commutative ring with unity $1 \neq 0$.

Definition. Two square matrices A and B over R are similar if there exists a nonsingular matrix P over R such that $A = P^{-1}BP$.

Similarity is an equivalence relation on the set $\mathcal{M}$ of $h \times h$ matrices and defines a partition on $\mathcal{M}$, the subsets of the partition being the equi-

This research was partially supported by National Science Foundation Grant No. GK-834.

valence classes of γ_m .

The state diagram (over J_m), without regard to specific relabeling of vertices, is an invariant under similarity transformations on matrix A since an isomorphic relabeling such that $\sigma' = P\sigma$ and $\omega' = P\omega$ for any nonsingular matrix P gives the affine transformation $\sigma'_{i+1} = PAP^{-1}\sigma_i + \omega'$ which has the same state diagram: $G(A, \omega) \approx G(PAP^{-1}, P\omega)$.

The objectives of this paper are twofold: (1) to determine a canonical form for similarity of matrices over R so each equivalence class contains a unique matrix whose form is especially convenient for determining the state diagram of the class, and (2) to establish an algorithm giving the state diagram $G(A, 0)$ of an arbitrary autonomous nonsingular LSC with matrix transformation A over J_m , where m is composite ($m = p_1^{n_1} p_2^{n_2} \dots p_r^{n_r}$). As will be seen, neither objective has been completely realized and further investigation is required.

The approach to be taken parallels to some extent that for vector spaces over finite fields, $J_p = GF(p)$. The method for finding state diagrams of affine transformations on these vector spaces is well known [3,4].

Since nonsingular transformations on finite dimensional modules over finite rings, by their nature, yield cycle sets, we have

Definition. The state diagram, $G(A, 0)$, of a nonsingular matrix A over J_m is described within an isomorphism by a cycle set $\Sigma = [\mu_0(1), \mu_1(K_1), \dots, \mu_p(K_p)]$ where the terms signify that the state diagram contains μ_i cycles of length K_i , for $i=1, \dots, p$.

Definition. Cycle terms $a_i = \mu_i(K_i)$ and $a_j = \mu_j(K_j)$ may be multiplied to give cycle term products of the form $\mu_i(K_i) \cdot \mu_j(K_j) = \mu_i \mu_j \gcd(K_i, K_j) (lcm[K_i, K_j])$.

Definition. The product of cycle sets $[a_1, \dots, a_s] \cdot [b_1, \dots, b_t]$ is the cycle set whose elements are all possible cycle term products $a_i b_j$, for $i=1, \dots, s$ and $j=1, \dots, t$.

Let $M[J_m]$ represent the module M over J_m . D. A. Bollman[5] has shown, for composite m , that $M[J_m]$ is isomorphic to a direct sum ($\oplus$) decomposition: $M[J_m] \approx M[J_{p_1 n_1}] \oplus \dots \oplus M[J_{p_r n_r}]$ and that the state diagram of a nonsingular matrix transformation on $M[J_m]$ is $\Sigma = \Sigma_1 \dots \Sigma_r$, a product of the cycle sets representing the state diagrams of the matrix transformation (reduced modulo $p_i^{n_i}$) on $M[J_{p_i n_i}]$, for $i=1, \dots, r$. Hence we may restrict our discussion to nonsingular transformations on modules over prime power modulus residue class rings: $M[J_{p^n}]$.

CANONICAL FORM FOR SIMILARITY

Although we are primarily interested in the decomposition of J_{p^n} -modules,

the following canonical form can be obtained as easily over any commutative ring (R) with unity $1 \neq 0$.

Cyclic Submodules

Let M be a finitely generated free (FGF) R-module and T be an endomorphism on M. $R[x]$ is the ring of polynomials over R in the indeterminate x.

Definition. A submodule $M' \subseteq M$ is said to be T-invariant if $TM' \subseteq M'$.

Definition. A submodule of M is T-cyclic, written $M(\sigma, T)$, if it is T-invariant and in addition is spanned by a given vector (σ) together with the vectors $g(T)\sigma$, for all $g(x) \in R[x]$. If $M(\sigma, T) = M$, σ is a cyclic vector for T.

Definition. The characteristic matrix of A over R is $xI - A$. The determinant of $xI - A$ is called the characteristic polynomial of A.

Definition. The null ideal ($\mathcal{N}$) of matrix A over R is the set of all polynomials $g(x) \in R[x]$ such that $g(A) = 0$. If the null ideal of A is principal, its unique monic generator is called the minimal polynomial of A. Since the null ideal is an invariant under similarity transformations, we may speak of the null ideal of T.

Definition. The T-annihilator of σ in M is the ideal $a(\sigma, T) \subseteq R[x]$ consisting of all $g(x) \in R[x]$ such that $g(T)\sigma = 0$.

If the T-annihilator of σ is a principal ideal, the unique monic generator $p_\sigma(x)$ of $a(\sigma, T)$ will also be called the T-annihilator of σ . The ideal $a(\sigma, T)$ contains the null ideal $\mathcal{N}$ of T since $\mathcal{N}\sigma = 0$, for all $\sigma \in M$. Notice that if $a(\sigma, T)$ is principal then the degree of $p_\sigma(x) > 0$, unless σ is the zero vector.

Theorem 1. Let σ be a non-zero vector in M and let $a(\sigma, T)$ be principal with monic generator $p_\sigma(x)$. Then

- The degree of $p_\sigma(x)$ equals the dimension of $M(\sigma, T)$,
- If the degree of $p_\sigma(x)$ is d, the vectors $\sigma, T\sigma, \dots, T^{d-1}\sigma$ form a basis for $M(\sigma, T)$,
- If T_σ represents the linear operator on $M(\sigma, T)$, induced by T, the minimal polynomial for T_σ is $p_\sigma(x)$.

Corollary 1.1. If M is a T-cyclic module then the minimal polynomial and the characteristic polynomial for T are identical.

Corollary 1.2. If M is a T-cyclic module generated by σ , then the $h \times h$ matrix of T in the ordered basis $B = \{\sigma, T\sigma, \dots, T^{h-1}\sigma\}$ is:

$$C_{p_\sigma(x)} = \begin{bmatrix} 0 & 0 & 0 & \cdot & 0 & -a_0 \\ 1 & 0 & 0 & \cdot & 0 & -a_1 \\ 0 & 1 & 0 & \cdot & 0 & -a_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & 1 & -a_{h-1} \end{bmatrix}$$

called the companion matrix of $p_\sigma(x) = x^h + a_{h-1}x^{h-1} + \dots + a_1x + a_0$.

Theorem 2. If T is a linear operator on the FGF R -module M , M is T -cyclic if and only if there is some ordered basis for M in which T is represented by the companion matrix of the minimal polynomial of T .

Corollary 2.1. If $C_{p(x)}$ is the companion matrix of a monic polynomial $p(x)$, then $p(x)$ is both minimal polynomial and characteristic polynomial of $C_{p(x)}$.

Rational Decomposition of R -modules

Let T be a linear transformation on the FGF R -module M . Under what conditions are there vectors $\sigma_1, \dots, \sigma_w$ such that $M = M(\sigma_1, T) \oplus \dots \oplus M(\sigma_w, T)$? If such a module decomposition occurs, T induces (by restriction) on each submodule $M(\sigma_i, T)$ a linear transformation T_i , and we may say that T is a direct sum of the operators $T_1, \dots, T_w$: $T = T_1 \oplus \dots \oplus T_w$. An ordered basis B_i can be chosen for each $M(\sigma_i, T)$ with corresponding A_i matrix representation for T_i and $B = \bigcup_{i=1}^w B_i$ then forms a basis for M , giving

$$A = A_1 \oplus \dots \oplus A_w = \begin{bmatrix} A_1 & 0 & . & 0 \\ 0 & A_2 & . & 0 \\ . & . & . & . \\ 0 & 0 & . & A_w \end{bmatrix}$$

The conditions for this type of decomposition are given by the following theorem. However, a more desirable form for the conditions will be derived later in terms of determinantal ideals and invariant factors.

Theorem 3. Let T be a linear transformation on the FGF R -module M with the dimension of M greater than 1. If the T -annihilators of M are principal, i.e., $a(\sigma_j, T) = (p_j(x))$ for all σ_j in M , then there exist non-zero vectors $\sigma_1, \dots, \sigma_w$ in M such that

a. $M = M(\sigma_1, T) \oplus \dots \oplus M(\sigma_w, T)$

b. $p_{i+1}(x)$ divides $p_i(x)$ for $1 \leq i \leq w-1$

Furthermore, the integer w and the T -annihilators are uniquely determined by (a) and (b) and the fact that no $\sigma_i = 0$.

Corollary 3.1. If $\sigma_1, \dots, \sigma_w$ satisfy the conditions of the theorem, then $a(\sigma_1, T) = (p_1(x))$ is the null ideal of T and $p_1(x)$ is its minimal polynomial.

By Corollary 1.1, the restrictions of T to the T -cyclic submodules $M(\sigma_1, T), \dots, M(\sigma_w, T)$, i.e., $T_1, \dots, T_w$, have cyclic vectors $\sigma_1, \dots, \sigma_w$, and for any T_i , $i=1, \dots, w$, $p_i(x)$ is both the characteristic polynomial and the minimal polynomial. Then, from the direct sum of T -cyclic submodules:

Corollary 3.2. The characteristic polynomial for T is $f(x) = p_1(x) \dots p_w(x)$.

Assume principal T -annihilators $(p_i(x))$, $i=1, \dots, w$. Let the cyclic ordered basis for $M(\sigma_i, T)$ be $B_i = \{\sigma_i, \dots, T^{h_i-1}\sigma_i\}$ where h_i is the dimension of $M(\sigma_i, T)$ and the degree of the annihilator $p_i(x)$. The matrix representing

the induced operator T_i in this basis is the companion matrix $C_{p_i(x)}$. Then the matrix of T with the ordered basis $B = \bigcup_{i=1}^w B_i$ is the rational matrix:

$$C_T = \begin{bmatrix} C_{p_1(x)} & 0 & 0 \\ 0 & C_{p_2(x)} & 0 \\ & & \ddots \\ 0 & 0 & C_{p_w(x)} \end{bmatrix}$$

Theorem 4. Let A be an $h' \times h'$ matrix over R representing a linear transformation T on the FGF R -module M . If T allows a direct sum decomposition of the form: $M = M(\sigma_1, T) \oplus \dots \oplus M(\sigma_w, T)$, with $\sigma_1, \dots, \sigma_w$ in M , then A is similar to one and only one matrix C_A (or C_T) in rational form.

Determinantal Ideals

Let A be an $h' \times h'$ matrix over R representing a linear transformation T on a FGF R -module M .

Definition. The determinantal ideals (D_i) of the characteristic matrix $xI-A$ are the $h'+1$ ideals $D_0, D_1, \dots, D_{h'}$, where D_i is generated by the non-zero determinants of all possible $i \times i$ submatrices of $xI-A$, $i=1, \dots, h'$, and $D_0=(1)$.

Definition. If the i^{th} determinantal ideal of $xI-A$, is principal, then its monic generator $d_i(x)$ is called the i^{th} determinantal divisor of $xI-A$.

Theorem 5. Let A be an $h' \times h'$ matrix over R . The i^{th} determinantal ideal of $xI-A$ contains the next determinantal ideal, i.e., $D_{i+1} \subseteq D_i$, $0 \leq i < h'$.

Corollary 5.1. When D_i and D_{i+1} , for $0 \leq i < h'$, are principal ideals, then $d_i(x)$ divides $d_{i+1}(x)$.

Invariant Factors

Assume that each determinantal ideal of $xI-A$, for given A over R , is principal.

Definition. The invariant factors of $xI-A$, for $i=1, \dots, h'$, are the quotients $\alpha_i(x) = d_i(x)/d_{i-1}(x)$.

Obviously, all the invariant factors exist if and only if all the determinantal ideals are principal.

Theorem 6. The determinantal ideals of the characteristic matrix of A over R are invariants under similarity transformations on A .

Theorem 7. If the invariant factors of the characteristic matrix of A over R exist, they are a complete set of invariants under similarity transformations on A .

Theorem 8. If T is a linear transformation on the h' -dimensional R -module M , with characteristic polynomial $f(x) = p_1(x) \dots p_w(x)$, and if T allows direct sum decomposition into T -cyclic submodules, then the invariant factors of T with degree greater than zero are the polynomials $p_1(x), \dots, p_w(x)$ in reverse

order, that is, $p_1(x) = \alpha_{h^1}(x), \dots, p_w(x) = \alpha_{h^1, -w+1}(x)$.

If $h^1 = w$, $\alpha_1(x) = d_1(x) = p_w(x)$. Otherwise, $\alpha_1(x) = d_1(x) = 1$.

Primary Decomposition of a J_{p^n} -module

Definition. Given polynomials $f(x)$ and $g(x)$ of positive degree in $R[x]$, $f(x)$ is related to $g(x)$ if there exists a polynomial $h(x) \in R[x]$ but not in $(g(x))$ such that $h(x)f(x) \equiv 0 \pmod{g(x)}$.

A useful test for related polynomials can be adapted from McCoy[6] or Lang[7], as follows:

Theorem 9. If $f(x)$ is a monic polynomial of positive degree, the non-zero polynomial $g(x) \in R[x]$ is related to $f(x)$ if and only if the resultant $R(f, g)$ is a divisor of zero in R .

Definition. A polynomial $f(x)$ in $R[x]$ is primary if it is the associate of a power of a monic irreducible polynomial $q(x)$, i.e., $f(x) = c[q(x)]^e$ where c is a unit in R and e is a positive integer.

The remainder of module decomposition will be developed over prime power modulus residue class rings because of the restrictions made by the following Lemma. The Lemma is essential for proving theorems 10 and 11 since $J_{p^n}[x]$ is not a unique factorization domain.

Lemma. Let $f(x)$ in $J_{p^n}[x]$ be monic with monic primary factors such that $f(x) = f_1(x)f_2(x) = f_1^i(x)f_2^j(x)$ where $f_1(x) \neq f_1^i(x)$ and $f_1(x) \neq f_2^j(x)$. Then $f_1(x)$, $f_2(x)$, $f_1^i(x)$, and $f_2^j(x)$ are related. Conversely, if $f(x)$ in $J_{p^n}[x]$ has two unrelated monic primary factors, the factorization in monic primary factors is unique.

Definition. The elementary divisors of the linear transformation T on the T -cyclic J_{p^n} -module M are the products of related primary polynomial factors of the minimal polynomial of T . The set of elementary divisors of T on a rationally decomposable J_{p^n} -module M is the combination of the sets of elementary divisors of the T -cyclic submodules of M , counted according to their multiplicities.

If R is a field, a polynomial $f(x)$ can be related to another polynomial $g(x)$ if and only if $f(x)$ and $g(x)$ in $R[x]$ have a common factor. Hence, two primary polynomials are related if and only if they are powers of the same irreducible polynomial over the field. For a finite field, J_p , therefore, the above definition of elementary divisors coincides with that commonly in use by MacLane and Birkhoff[8], Thrall and Tornheim[9], Curtis and Reiner[10] and others.

Theorem 10. Let T be a linear transformation on the T -cyclic FGF J_{p^n} -module $M(\sigma, T)$ with minimal polynomial $m(x) = m_1(x) \dots m_v(x)$, factorable to v elementary divisors. Then $M(\sigma, T)$ can be decomposed into a direct sum of T -cyclic

submodules which are the kernels of $m_i(T)$, $i=1, \dots, v$, operating on $M(\sigma, T)$:
 $M(\sigma, T) = \text{KER } m_1(T) \oplus \dots \oplus \text{KER } m_v(T)$.

Theorem 11. The elementary divisors of a linear transformation T on a FGF T -cyclic J_{p^n} -module are unique.

Theorem 12. Let the matrix A represent a linear transformation T on a FGF J_{p^n} -module M which allows a rational decomposition to T -cyclic submodules. Then the elementary divisors of T are a complete set of invariants under similarity transformations on A .

Corollary 12.1. Any matrix A , representing a linear transformation T on a J_{p^n} -module for which invariant factors exist, is similar to one and only one direct sum of companion matrices belonging to the elementary divisors of T . This matrix is called the primary rational canonical form of A (or T):

$$C_m(x) = \begin{bmatrix} C_{m_1}(x) & 0 & 0 \\ 0 & C_{m_2}(x) & 0 \\ 0 & 0 & C_{m_v}(x) \end{bmatrix}$$

Each submatrix is the companion matrix $C_{m_i}(x)$ of the elementary divisor $m_i(x) = x^{h_i} + a_{h_i-1}x^{h_i-1} + \dots + a_1x + a_0$.

Corollary 12.2. Any matrix A over J_{p^n} has a primary rational canonical form if (but not only if) each of the determinantal ideals of $xI-A$ is principal.

This last corollary gives sufficient conditions for a primary rational decomposition in simple and useful form.

STATE DIAGRAMS FOR DIRECT SUM DECOMPOSITIONS OF J_m -MODULES

Product of Cycle Sets for Direct Sums

Theorem 13. Let A be a nonsingular matrix representing linear transformation T on the J_m -module M . With each T -invariant submodule M_i of M there can be associated a cycle set describing the state diagram of A_i , the restriction of A to the submodule M_i . If M is the direct sum of T -invariant submodules $M = M_1 \oplus \dots \oplus M_t$, then the cycle set of $G(A, 0)$ is the product of cycle sets $G(A_1, 0), \dots, G(A_t, 0)$.

As mentioned, a J_m -module is isomorphic to a direct sum of J_{p^n} -modules, $M[J_m] \approx M[J_{p_1 n_1}] \oplus \dots \oplus M[J_{p_r n_r}]$, where T induces on each module a restricted linear transformation $T = T[J_m] \approx T[J_{p_1 n_1}] \oplus \dots \oplus T[J_{p_r n_r}]$. If each $T[J_{p_i n_i}]$, $i=1, \dots, r$, allows a primary rational canonical decomposition over the respective residue class ring, then M is isomorphic to the direct sum:

$$M \approx M_{m_{11}}(x)[J_{p_1 n_1}] \oplus \dots \oplus M_{m_{1v(1)}}(x)[J_{p_1 n_1}] \oplus \dots \oplus M_{m_{rv(r)}}(x)[J_{p_r n_r}].$$

The cycle set for each term of the sum can be calculated with the aid of the corresponding elementary matrix. Theorem 13 gives the state diagram of the

J_m -module as the product of the $v(1)+v(2)+\dots+v(r)$ cycle sets associated with the terms in the direct sum.

Cycle Sets of Nonsingular Elementary Matrix Transformations on J_{p^n} -modules

The cycle set of a nonsingular elementary matrix, $C_e = C_{m_i(x)}[J_{p^n}]$, representing transformation T on the FGF J_{p^n} -module M , may be obtained from the minimal polynomial of C_e . However, the determination of $G(C_e, 0)$ over J_{p^n} when $n > 1$ is more complex than when $n = 1$, $J_p = GF(p)$, for two reasons: (1) the elementary divisor for C_e over J_{p^n} , $n > 1$, has been defined in a more general way so as to include all related primary polynomials that occur in an invariant factor of the characteristic matrix $xI - A$; (2) the cycle structure for C_e over J_{p^n} , $n > 1$, is determined by the ascending sequence of T -invariant submodules within M . These submodules are induced by the ascending sequence of principal ideals within J_{p^n} : $(p^{n-1}) \subset (p^{n-2}) \subset \dots \subset (p)$, where (p) is the maximal ideal and the radical of the local ring J_{p^n} .

Period of Vectors in J_{p^n} -modules

Definition. The period of a nonsingular matrix A over J_{p^n} is the least positive integer k such that $A^k = I$.

Methods for determining the periods of matrices over J_{p^n} have been developed by D. A. Bollman [11].

Definition. The period of a given vector $(\sigma \in M)$ under the nonsingular transformation represented by matrix A is the least positive integer $k(\sigma)$ such that $A^{k(\sigma)}\sigma = \sigma$.

The period of σ is the length of the cycle including σ . Clearly, $k(\sigma) | k$.

Definition. The period (or the exponent) of a polynomial $f(x)$ is the least positive integer t such that $f(x) | (x^t - 1)$.

The period of a nonsingular matrix A equals the period of the minimal polynomial $m(x)$ of the matrix A .

Definition. A J_m -module M is said to be of maximal period under linear transformation T if every non-zero vector in M has period equal to $m^h - 1$, where h is the order of matrix A representing T .

D. A. Bollman [5] has shown that, when m is not prime, there is no J_m -module under any linear transformation that is of maximal period.

The isomorphism $\beta: M \rightarrow J_{p^n}[\xi]$, where $J_{p^n}[\xi]$ is the ring of polynomials modulo $m(x)$ over J_{p^n} , simplifies the following theorems and their proofs.

$$\sigma = \begin{bmatrix} a_0 \\ \cdot \\ \cdot \\ a_{h-1} \end{bmatrix} \mapsto \sigma(\xi) = a_{h-1}\xi^{h-1} + \dots + a_0, \quad a_i \in J_{p^n}$$

Theorem 14. If the linear transformation T is represented by C_e and σ in M is represented by $\sigma(\xi)$ in $J_{p^n}[\xi]$, then the p^{th} successor of σ under T is

represented by $\xi^p \sigma(\xi)$.

Corollary 14.1. The period of any vector $\sigma \in M$ over J_{p^n} is the least positive integer $k(\sigma)$ such that $\xi^{k(\sigma)} \sigma(\xi) = \sigma(\xi)$ or such that $(1 - \xi^{k(\sigma)}) \sigma(\xi) = 0$.

Definition. The modulo- p^i polynomial $f^i(x)$ is the polynomial formed by replacing the coefficients of $f(x)$ over J_{p^n} by their (p^i) -residues for $i < n$.

Definition. The period k_i of the modulo- p^i polynomial $f^i(x)$ is the least positive integer t such that $f^i(x) | (x^t - 1) \bmod p^i$.

Theorem 15. Let $m(x)$ be the minimal polynomial of the nonsingular $h \times h$ matrix C_e operating on the J_{p^n} -module of h -tuples. If $\sigma(\xi) \in J_{p^n}[\xi]$ modulo $m(x)$ and $\sigma(\xi)$ is a multiple of p^{n-i} , $1 \leq i \leq n$, then the period $k(\sigma)$ of the corresponding σ divides the period k_i of $m^i(x)$: $k(\sigma) | k_i$.

Theorem 16. Given $m(x)$ and C_e as in the previous theorem, if (1) $\sigma(\xi) \in J_{p^n}[\xi]$ is a multiple of p^{n-i} , $1 \leq i \leq n$, but not a multiple of p^{n-i+1} and (2) the polynomial $m^i(x)$ is irreducible mod p^i , then the period of σ is equal to the period of $m^i(x)$: $k(\sigma) = k_i$.

If $m^i(x)$ is irreducible, every $\sigma(\xi)$ in $(p^{n-i})J_{p^n}[\xi]$ which is not in $(p^{n-i+1})J_{p^n}[\xi]$ has the same cycle length k_i .

Cycle Sets in J_{p^n} -modules under Nonsingular Transformations with Multiply Irreducible Minimal Polynomials

Theorem 17. The J_{p^n} -module M of h -tuples has $p^{ih} - p^{(i-1)h}$, $1 \leq i \leq n$, distinct vectors $\sigma \in M$ such that $\sigma(\xi)$ is a multiple of p^{n-i} but not of p^{n-i+1} .

The polynomials $\sigma(\xi) \in J_{p^n}[\xi]$ modulo $m(x)$ that are multiples of p^{n-i} , for $i < n$, form a submodule (over J_{p^n}) which is induced by the ideal (p^{n-i}) , i.e., $(p^{n-i})J_{p^n}[\xi] \subset J_{p^n}[\xi]$, which implies that $(p^{n-i})M \subset M$.

Definition. A polynomial $f(x) \in J_{p^n}[x]$ is said to be multiply irreducible if it is irreducible mod p^i for $i=1, \dots, n$.

The following theorem is an immediate consequence of theorems 16 and 17.

Theorem 18. Let $m(x)$ be the minimal polynomial of the nonsingular $h \times h$ matrix C_e operating on a J_{p^n} -module M of h -tuples. If $m(x)$ is multiply irreducible then the state diagram $G(C_e, 0)$ is

$$\Sigma = [1(1), \frac{p^{h-1}-1}{k_1}(k_1), \frac{p^{h(p^h-1)}-1}{k_2}(k_2), \dots, \frac{p^{(i-1)h(p^h-1)}-1}{k_i}(k_i), \dots, \frac{p^{(n-1)h(p^h-1)}-1}{k_n}(k_n)]$$

Of course, k_1 is the period of $m^1(x) \bmod p$ and $k_n = k$ is the period of $m(x)$ and of C_e over J_{p^n} .

FURTHER RESEARCH

1. The primary rational canonical form for similarity is developed here for matrices, over the commutative ring R , whose characteristic matrices have principal determinantal ideals. A more general canonical form is needed to include all matrices over R . It is clear that nonsingular matrices

without associated principal determinantal ideals do partition into equivalence classes under similarity transformations and their state diagrams are composed of cycle sets. But a study of this area has not been made.

2. Nonsingular elementary matrices with reducible minimal polynomials have not been discussed here. While some results are available, a general method for obtaining the state diagrams of elementary nonsingular matrices from their minimal polynomials (with related factors) has not been found.

3. Nilpotent linear transformations on J_m -modules give tree structured state diagrams and singular transformations over J_m can be decomposed to a direct sum of nonsingular and nilpotent transformations. But algorithms for determining the state diagrams for these transformations do not now exist.

4. More general study would include affine transformations on J_m -modules. Such research is complete for vector spaces over finite fields [3,4].

ACKNOWLEDGMENT

The author would like to thank Dr. R. J. Leake for his encouragement.

REFERENCES

- [1] A. Gill, Linear Sequential Circuits, New York: McGraw-Hill, 1966.
- [2] D. A. Bollman, "On the periodicity of states in a linear sequential machine having composite modulus," Doctoral Dissertation, University of Illinois, Urbana, 1963.
- [3] A. Gill, "Graphs of affine transformations, with applications to sequential circuits," Seventh Annual Symposium on Switching and Automata, Berkeley, October 1966, pp. 127-135.
- [4] K.C. Wang, "Transition graphs of affine transformations on vector spaces over finite fields," *Journal of the Franklin Institute*, vol. 283, no. 1, January 1967, pp. 55-72.
- [5] D. A. Bollman, "Some periodicity properties of modules over the ring of polynomials with coefficients in a residue class ring," *SIAM Journal*, vol. 14, no. 2, March 1966, pp. 237-241.
- [6] N. H. McCoy, Rings and Ideals, Carus Mathematical Monograph No. 8, LaSalle, Illinois: Open Court, 1948.
- [7] S. Lang, Algebra, Reading, Mass: Addison-Wesley, 1965.
- [8] S. MacLane and G. Birkhoff, Algebra, New York: Macmillan, 1967.
- [9] R.M. Thrall and L. Tornheim, Vector Spaces and Matrices, New York: John Wiley, 1957.
- [10] C.W. Curtis and I. Reiner, Representation Theory of Finite Groups and Associative Algebras, New York: John Wiley, 1962.
- [11] D. A. Bollman, "Some periodicity properties of transformations on vector spaces over residue class rings," *SIAM Journal*, vol. 13, no. 3, September 1965, pp. 902-912.

SYNTHESIS OF STOCHASTIC SEQUENTIAL MACHINES

LARRY C. SCHOOLEY and GERRY L. KELLY

University of Arizona, Tucson, Arizona and University of Kansas,
Lawrence, Kansas.

ABSTRACT

Due to recent investigations by a number of researchers the mathematical structure of stochastic sequential machines is now fairly well understood. On the other hand, very little attention has been paid to the important areas of synthesis and application. This paper shows how stochastic sequential machines may be physically realized in terms of switching circuits, and outlines a number of important potential applications for such devices.

1. INTRODUCTION

A stochastic sequential machine (SSM), as defined by Carlyle (1) may be regarded as a mathematical model of stochastic finite-state systems, or as a generalization of deterministic sequential machines. As in the case of deterministic sequential machines, there are always finite sets of internal states, input symbols, and output symbols. Also, the operation proceeds in a sequential fashion, with new states and output symbols appearing at discrete intervals of time in response to a sequence of input symbols. In the case of stochastic sequential machines, however, the state transitions and appearance of output symbols are probabilistic in nature rather than deterministic. The model is quite general, as can be seen from the fact that the finite state communication channel (2), the additive noise machine of Eaton and Zadeh (3), the probabilistic automaton of Rabin (4) and Paz (5), and the deterministic sequential machines of Moore (6) and Mealy (7) can all be represented as special cases of an SSM.

The algebraic structure of such machines has been investigated by Carlyle (1,8), Bacon (9), Ott (10) and others who have derived input-output relationships, defined concepts of equivalence and dominance, and shown how internal states equivalent to other states or probability distributions on states can be eliminated. Due to these investigations the mathematical structure of such machines is now fairly well understood. On the other hand, very little effort has been made to answer the following very important questions:

1. What can stochastic sequential machines do that make them of special importance?
2. How can stochastic sequential machines be realized physically?

Regarding the first question it appears that there are a large number of important potential applications, some of which will be outlined later. Our primary purpose in this paper, however, is to establish a suitable method for synthesis of stochastic sequential machines. More specifically, we will show how any such machine can be constructed from switching circuits, and indicate how the design may be simplified in certain special cases which are of importance from the standpoint of applications.

2. DEFINITIONS

2.1 Stochastic Sequential Machine.

A stochastic sequential machine may be defined as a 4-tuple:

$$SSM = (X, Y, S, M)$$

where

$X = (x_1, x_2, \dots, x_p)$ is a finite set of p distinct input symbols.

$Y = (y_1, y_2, \dots, y_q)$ is a finite set of q distinct output symbols.

$S = (s_1, s_2, \dots, s_r)$ is a finite set of r internal states.

M is a $pr \times qr$ stochastic matrix whose entries are conditional probabilities which govern the behavior of the machine.

It is convenient to partition the matrix M into $r \times r$ submatrices and to identify these pq submatrices with the pairs (x_i, y_j) of input and output symbols in the natural order. The entries of each of these submatrices are identified in the natural order with pairs (s_k, s_l) from the set of states. Thus a specific entry in the M matrix might be written as $m_{kl}(y_j/x_i)$ and would be located in the k th row and l th column of the submatrix $M(y_j/x_i)$. This entry is the numerical value for the conditional probability that the output symbol is y_j and that the next state will be s_l , given that the input symbol is x_i and the present state is s_k . In a more concise notation, $m_{kl}(y_j/x_i) = p(y_j, s_l/s_k, x_i)$. Where the prime added to the state symbol s_l means "The next state will be s_l ," and the notation $p(\cdot/\cdot)$ denotes a conditional probability. Each row of the matrix is a probability vector which may be interpreted as defining a discrete random experiment with possible outcomes consisting of the qr mutually exclusive events spanning the $Y \times S$ space and with specified probabilities for each such outcome. Choosing a row by specifying input symbol and present state corresponds to choosing one of pr such random experiments.

This definition can be used as a mathematical description for several different kinds of mechanisms, differing in the sequence and timing of operation. For the purposes of the synthesis described herein, the operating sequence is assumed to be as follows. Inputs appear at regular time intervals $(0, T, 2T, \dots)$ and remain through the period of length T . As the inputs appear, the machine switches from the previous state to

the present state, the output appears, and the next state is generated and stored in memory. The machine then remains in this configuration until the next time period when another input appears and the process repeats. For purposes of this exposition the switching may be considered to take place instantaneously although of course in actual devices finite time delays occur and some sort of gating or clock system may be required.

2.2 Switching Circuits and Discrete Probability.

The fundamental relations involving discrete probability and switching theory were presented by Warfield (11). The occurrence of an event is represented by the closure of a switch. The probability of occurrence of an event is represented by the probability of closure of a switch.

2.3 Binary White Source.

A binary white source is defined to be a history-independent sequential source of binary digits with equal probability for appearance of zeros and ones. Such sources can be obtained by sampling operations on natural fluctuation phenomena, or approximated by pseudo-random means. For example, one can take a non-biased random process like resistance noise, sample it, let the output be 1 when the signal is positive and zero when the signal is negative, and get a very good approximation to a source with the stochastic matrix

$$\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$$

In other words, one can produce a single binary white source. Then with a duplicate setup one can produce another such source, independent of the first one. Continuing in this way, one can develop any reasonable number of independent binary white sources.

2.4 Stochastic Degree.

It will be required that all probabilities be approximated as finite binary fractions. If a probability vector is approximated such that the largest number of binary places required to represent any entry is d , then the vector is said to be of stochastic degree d . Similar definitions apply to rows of stochastic matrices, or to the matrices themselves.

2.5 Random Experiment Networks.

Warfield has shown (12) how any set of switches with a specified distribution of the probabilities of closure can be realized from binary white sources and combinational logic. In particular, he has shown how a complete set of n mutually exclusive events with specified probability distribution can be so realized. If the largest stochastic degree of any of the probabilities is d , then d independent binary white sources are required. We call such a set a random experiment network of order n and stochastic degree d . It may be pictured as a black box with n accessible switch contacts.* At any instant of time one and only one of the switch contacts is closed, and the probability of contact i being the one which is closed is the probability p_i which was specified for the occurrence of that event.

3. SYNTHESIS PROCEDURES

3.1 A Canonical Scheme

Theorem: Any stochastic sequential machine with r internal states, and with an M matrix of stochastic degree d , may be realized by $\langle \log_2 r \rangle$ ** binary memory units, d binary white sources, and combinational logic. **Proof:** The proof is a constructive proof in that a machine of this class is designed. First let us review a method for construction of deterministic sequential machines as shown in Figure 1. Here $U = (u_1, u_2, \dots, u_a)$, $V = (v_1, v_2, \dots, v_b)$ and $Z = (z_1, z_2, \dots, z_c)$ are all sets of binary variables which are used with some code to represent the input symbols, state set, and output symbols. If there are to be p input symbols, q output symbols, and r internal states, $a = \langle \log_2 p \rangle$, $b = \langle \log_2 r \rangle$, and $c = \langle \log_2 q \rangle$. The machine is entirely deterministic. Given any input x_i determined by a particular value of U and any present state s_k determined by the values of V , a next state s'_1 (corresponding to variable values V') is generated by combinational logic and stored in the memory, and the output Z is generated by combinations of U and V . After time T , the next state becomes the present state, a new input appears, and the process repeats. Design procedures for such machines are well described in a number of books.

The canonic construction of the stochastic sequential machine will be similarly done. Instead of one combinational logic network to generate next state from present state and input, there will be $p \cdot r$ networks. Let them be denoted B_{ikl} . The inputs to these networks correspond to the internal variables V and the machine input variables U and the outputs correspond to the internal variables V . They are constructed such that if the input (present state and machine input) corresponds to s_k and x_i

* It is convenient to describe our SSM synthesis in terms of switch contacts being open or closed but of course any of a wide variety of physical devices may be used in actual realizations.

** $\langle \log_2 z \rangle$ is defined to be the smallest integer n such that $2^n \leq z$.

then the output (next state) corresponds to s_j , for all i, k , and l . Instead of one combinational logic network to generate output from present state and input, there will be pqr such networks. Let them be denoted C_{ikl} . The inputs to these networks correspond to the internal variables V and machine input variables U and the outputs correspond to the machine output variables Z . They are constructed such that if the present state is s_i and the input symbol is x_i then the output corresponds to y_j , for all i, j , and k .

One also needs a combinational network whose output consists of pr mutually exclusive indicators. The inputs to this network are the present state variables and the machine input variables, and one and only one of the pr indicators is one (closed switch) depending on the input symbol and present state. Call this network D and let it have pr outputs (switch contacts) identified by $D(i, k)$. If the input is x_i and the present state is s_k then contact $D(i, k)$ is closed and all others are open.

Finally one needs pr random experiment networks. Call these A_{ik} . They are each of order qr and each one corresponds to one row of the probability matrix M . If the largest stochastic degree of any entry in the M matrix is d , then any of these can be realized with d binary white sources and combinational logic. Since only one of these networks will be in use at any given time, the same d white sources can be used to drive all pr of the A_{ik} networks.

A block diagram of the machine is shown in Figure 2. In this figure the heavy lines carry binary variables representing input, present state, output, and next state. The light lines represent indicators (closed switch contacts) used to select proper networks. Operation of the machine may now be described as follows. Given a present state and an input at time $t = nT$, network D is used to select one (and only one) of the random experiment networks. If input is x_i and present state is s_k then contact $D(i, k)$ is closed and network A_{ik} (which corresponds to the proper row of the probability matrix M) is selected. A random experiment with prescribed probabilities is thus performed and the output of network A_{ik} (contact which is closed) identifies the next state (say s_j) and the output (say y_j). This contact (along with the contact $D(i, k)$) is used to select networks B_{ikl} and C_{ikl} . At this point the network C_{ikl} which has been selected provides the proper output y_j . The network B_{ikl} dictates that the input to the memory unit will be state l , and thus at the next interval (time $t = (n + 1)T$) the internal state will shift to the proper next state s_l .

The machine now remains in this condition until time $t = (n + 1)T$ when the state transition occurs, the next input appears, and the process is repeated. All transitions thus occur in accordance with the probabilities given by the matrix M and the synthesized machine behaves as desired.

3.2 Some Simplifications.

The constructive proof of the previous section shows a canonic scheme for constructing any stochastic sequential machine. Even in the general case this construction can be simplified in a manner which greatly reduces the amount of combinational logic required. Rather than pr^2 networks to generate next state from present state and input, there are r networks

which generate the r possible states directly. These can be denoted B_j . Also, rather than pqr networks to generate output from present state and input, there are q networks which generate the q possible output symbols directly. These can be denoted C_j . One still needs the combinational network D , consisting of pr mutually exclusive indicators whose output (closed contact) signifies position in the cartesian product space $X \times S$.

Finally, one still needs pr random experiment networks. Call these A_{ik} . They are each of order pr and stochastic degree d , and each one corresponds to one row of the probability matrix M . As before, if the largest stochastic degree of any entry in the M matrix is d , then any of these can be realized by d white sources and combinational logic. Since only one will be in use at any given time, the same d white sources can be used to drive all pr of the A_{ik} networks. Operation of the machine may now be described as follows. Given a present state and an input at time $t = nT$, network D is used to select one (and only one) of the random experiment networks. If input is x_i and present state is s_j , then contact $D(i, k)$ is closed and network A_{ik} (which corresponds to the k proper row of the M matrix) is selected. A random experiment with prescribed probabilities is thus performed and the output of network A_{ik} (contact which is closed) identified the next state and the output. Say $A_{ik}(j, l)$ is closed at outcome of the experiment. Then this contact is used to select networks C_l and B_j . At this point the network C_l which has been selected provides the proper output y , and the network B_j dictates that the input to the memory unit is state s_l . Thus at the next time interval ($t = (n + 1)T$) the internal state will shift to the proper next state s_l . The machine now remains in this condition until time $t = (n + 1)T$ when the state transition occurs, the next input appears, and the process is repeated. All transitions occur in accordance with the probabilities given by matrix M and the synthesized machine behaves as desired.

Considerable additional reduction in complexity is possible in certain special cases which are of importance from the standpoint of application. These include the cases where the probability distributions for output and next state are independent, and where either the output function or the next state function is deterministic. Details may be found in the work by Schooley (13). In addition, procedures are given for constructing any stochastic sequential machine as an equivalent machine whose output is a deterministic function of the present state. This is important for avoiding timing problems in series-parallel connections of machines. The design of stochastic sequential sources, which are stochastic sequential machines with no inputs, is also described.

3.3 Provision of Initial Conditions.

The simplified synthesis of the previous section also suggests a simple way to provide for the initial condition of the machine. In the mathematical model, the initial condition is given formally by specifying an initial probability distribution on the state space. For the physical implementation, we need merely construct a random experiment network of order r in accordance with these probabilities. If the stochastic degree of the initial distribution vector is less than or equal to that of the M matrix, then the same d white sources used in the main machine can be used for this network. If it is greater, some additional white sources will be required. This network is used to charge the memory banks with a state vector according to the specified initial state probability distribution. Then when the first clock pulse and input signal appear, the memory shifts and the state selected by this random experiment becomes the present state associated with the first input symbol. Operation then proceeds as previously described.

4. APPLICATIONS

Recent work (13) establishes the existence of a wide variety of important potential applications of stochastic sequential machines. This is done by showing how a number of sample problems from various fields can be put in one to one correspondence with the stochastic sequential machine model, and then referencing related literature. The areas described are application to numerical calculations, including matrix multiplication and inversion and solution of partial differential equations; simulation of naturally occurring random processes, including problems from statistical communication theory, queueing theory, and the social and biological sciences; optimal control of stochastic systems, including pursuit problems, game theory problems, and business management; and applications in teaching and learning, including design of learning machines, pattern recognition devices, and adaptive control systems. The key to feasibility in all of these applications is speed, and this in turn depends primarily on the speed with which the binary white sources can be made to operate.

5. CONCLUSIONS

Any stochastic sequential machine can be realized physically from combinational logic, binary memory elements, and binary white sources. The amount of combinational logic required is quite large, for general machines of appreciable size, but appears to be within the current state of the art, particularly in view of the recent advantages in integrated circuit technology. Also, some of the important special cases require much less hardware for realization. A number of significant potential applications have been identified and it appears probable that these machines or others like them will eventually be constructed and used. This must, however, await further research, particularly the development of high speed binary white sources along with means for verifying their statistics, and further details of economical logic design.

REFERENCES

1. J. W. Carlyle, "Reduced Forms for Stochastic Sequential Machines," J. Math. Anal. Appl., vol. 7, pp. 167-175, 1963.
2. R. Ash, Information Theory, New York: Interscience Publishers, Wiley, 1965.
3. J. H. Eaton and L. A. Zadeh, "Optimal Pursuit Strategies in Discrete-State Probabilistic Systems," Journal of Basic Engineering, Trans. of the A.S.M.E., series D, vol. 84, pp. 23-29, 1962.
4. M. O. Rabin, "Probabilistic Automata," Information and Control, vol. 6, pp. 230-245, 1963.
5. A. Paz, "Some Aspects of Probabilistic Automata," Information and Control, vol. 9, pp. 26-60, 1966.
6. E. F. Moore, "Gedanken-experiments on Sequential Machines," in Automata Studies, Princeton Univ. Press, Princeton, N.J., 1956.
7. G. H. Mealy, "A Method for Synthesizing Sequential Circuits," Bell Sys. Tech. J., vol. 34, pp. 1045-1079, 1955.

8. J. W. Carlyle, "State-Calculable Stochastic Sequential Machines, Equivalences and Events," 1965 IEEE Conf. Record on Switching Circuit Theory and Logical Design, pp. 258-263, 1965.
9. G. C. Bacon, "Minimal-State Stochastic Finite State Systems," IEEE Trans. on Circuit Theory, vol. CT-11, pp. 307-308, 1964.
10. G. H. Ott, "Reconsider the State Minimization Problem for Stochastic Finite State Systems," IEEE Conf. Record of Seventh Ann. Symposium on Switching and Automata Theory, pp. 267-273, 1966.
11. J. N. Warfield, "Switching Circuits as Topological Models in Discrete Probability Theory," IRE Trans. on Electronic Computers, vol. EC-7, 1958.
12. J. N. Warfield, Principles of Logic Design, Boston: Ginn and Co., 1963, Ch. 16.
13. L. C. Schooley, "Synthesis and Application of Stochastic Sequential Machines," Technical Report 149-1, Center for Research in Engineering Science, University of Kansas, Lawrence, July, 1968.

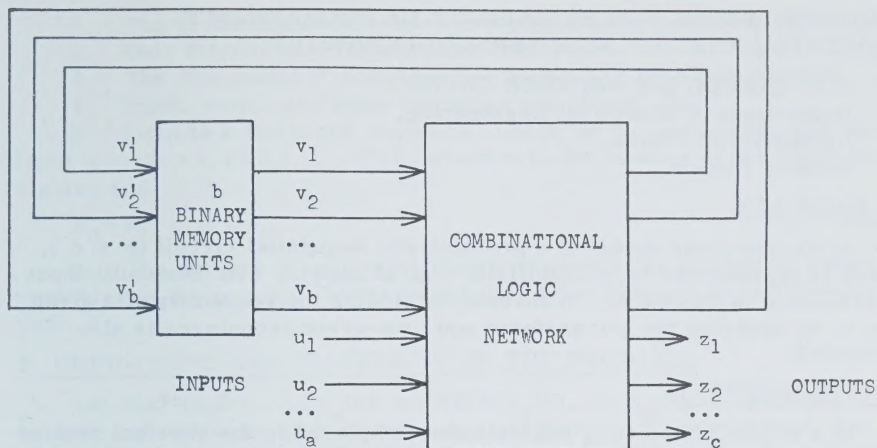


Fig. 1 Deterministic Sequential Machine

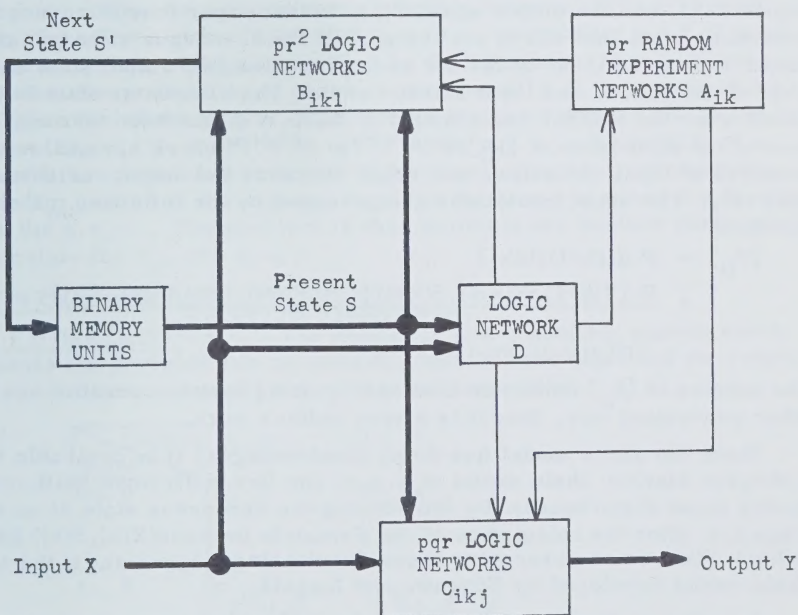


Fig. 2 Stochastic Sequential Machine

A MARKOV MODEL FOR RELIABILITY OF PROBABILISTIC SEQUENTIAL CIRCUIT WITH BERNOULLI INPUTS*.

G. S. GLINSKI and WHO-KEE CHUNG
Department of Electrical Engineering,
University of Ottawa,
Ottawa, Canada.

1. ABSTRACT

A Markov chain model of a probabilistic sequential circuit (p. s. c.), which is represented by probabilistic logical matrix, with Bernoulli input is presented in this paper. A method for finding the reliability of a given p. s. c. by applying the z-transform and flow-graph techniques is also presented.

2. INTRODUCTION

In a sequential circuit, each element employed in the physical realization may be associated with probability of erroneous operation. The use of probabilistic logical matrix (P-matrix, $[P]$) to estimate the reliability of combinatorial circuit was introduced by Udagawa, Inagaki^[1] and Levin^[2]. In 1964 Udagawa and Inagaki^[3] extended their previous work and introduced the idea of total state transition matrix, $[P_\Omega]$, to estimate the reliability of p. s. c., Fig. 1, using Markov chain model with Markov input without noise disturbance. The states of $[P_\Omega]$, $\Omega(k)$; $k = 0, 1, \dots$ are the direct product (Kronecker product) of the input state, $X(k)$, the state of the unit delay elements, $S(k)$, and the output state, $Y(k)$. In that paper they introduced the erroneous state into which are lumped all the erroneous state transitions, output states and input states for each present state, output state and input state of the p. s. c. and input Markov chain. The erroneous state is introduced after the Markov chain model of the p. s. c. had been formed. For this reason the dimension of $[P_\Omega]$ is $2^{n+r+m} \times 2^{n+r+m}$ where n, r and m are the numbers of input variables, unit delay elements and output variables respectively. The state transitions are governed by the following matrix equation:

$$\begin{aligned} [P_\Omega] &= P \{ \Omega(k+1) / \Omega(k) \} \\ &= \frac{P \{ Y(k+1) / X(k+1), S(k+1) \} \cdot P \{ S(k+1), Y(k) / X(k), S(k) \}}{P \{ Y(k) / X(k), S(k) \}} \\ &\quad \cdot P \{ X(k+1) / X(k) \} \end{aligned} \quad (1)$$

The entries of $[P_\Omega]$ cannot be obtained by using matrix operations or by any other convenient way, thus it is a very tedious work.

Since the above model has these disadvantages, it is desirable to have a simpler Markov chain model of p. s. c. for Bernoulli input (with or without binary noise disturbance). By introducing the erroneous state at an earlier stage, i. e. after the calculation of the P-matrix between $(X(k), S(k))$ and $(S(k+1), Y(k))$, we can overcome these disadvantages appearing in the Markov chain model developed by Udagawa and Inagaki.

In order to simplify the problem the following assumptions are made:

* This work was supported in part by the National Research Council of Canada, Grant No. A-875.

1. The probabilities of errors are statistically independent.
2. Only intermittent failures are considered.
3. The component P-matrices are known and time independent.
4. Input, output and state variables are binary $\{0, 1\}$.
5. Input is a Bernoulli sequence. Let p_i be the probability that the input n-tuple = i , $i = 0, 1, \dots, 2^n - 1$, where n is the number of the input variables and

$$p_0 + p_1 + \dots + p_{2^n - 1} = 1$$

6. The number of unit delay elements equals the number of state variables of the p. s. c.. There is one unit delay element in each feedback path.

3. DEFINITIONS AND STATEMENT OF THE PROBLEM

Let $X(k) = (x_1(k), \dots, x_n(k))$ and $Y(k) = (y_1(k), \dots, y_m(k))$ be the input n-tuple and output m-tuple of the p. s. c. at time $t = k$ respectively. Let $S(k) = (s_1(k), \dots, s_r(k))$ and $S(k+1) = (s_1(k+1), \dots, s_r(k+1))$ be the state r-tuples of the p. s. c. at time $t = k$ and $(k+1)$ respectively. By the definition of sequential circuit (Mealy model) [4],

$$S(k+1) = F(X(k), S(k)) \quad (2)$$

$$Y(k) = G(X(k), S(k)) \quad (3)$$

The block diagram of a sequential circuit is shown in Fig. 1. The D's are unit delay elements.

In p. s. c. there is a one to many mapping between $(X(k), S(k))$ and $(S(k+1), Y(k))$. For a given $(X(k), S(k))$ there is a probability $p \cong 1$ of producing correct $(S(k+1), Y(k))$ as specified by the equations (2) and (3) (we assume that the probability of error is very small). If the result of the actual operation of the p. s. c. differs from the results provided by the equations (2) and (3), there is an error in the p. s. c.. Reliability of a p. s. c. will be considered in terms of the mean-time-to-first-system-failure, T_m , which is the average time for the error to occur for the first time in the p. s. c.. The problem is thus to obtain the Markov chain model and calculate the T_m of a p. s. c..

4. NOTATION AND BASIC PROPERTIES

4.1 P-MATRIX [1] [5]. The P-matrix is a transition matrix which represents the probabilistic relationship between the input and the output of a given switching circuit. It has several properties.

(i)

Output Input	$0 \dots 0 \quad \dots \quad 1 \dots 1$
$[P] =$	$\begin{bmatrix} p_{0,0} & \dots & p_{0,2^w-1} \\ \vdots & & \vdots \\ p_{2^q-1,0} & \dots & p_{2^q-1,2^w-1} \end{bmatrix}$

where w is the total number of the output and/or the state variables and q is the total number of the input and/or the state variables.

(ii)

$$0 \leq p_{i,j} \leq 1, \quad i = 0, 1, \dots, 2^q - 1; \quad j = 0, 1, \dots, 2^w - 1.$$

$$\sum_{j=1}^{2^w-1} p_{i,j} = 1, \quad \forall i$$

(iii)

$$\sum_{i=0}^{2^q-1} \sum_{j=0}^{2^w-1} p_i \cdot p_{i,j} = 1;$$

4.2 SERIES COMBINATION OF CIRCUITS^{[1][5]}. The P-matrix of series combination of N circuits as shown in Fig. 2 has n input variables and m output variables and is given by

$$[P] = [P_1] \cdot [P_2], \dots, [P_N]$$

provided that the number of output variables of (i-1)th logical element equals the number of input variables of i-th logical element $i = 2, 3, \dots, N$. The operator "." is the usual matrix multiplication.

4.3 "SIDE BY SIDE" (JUXTAPOSING) COMBINATION OF CIRCUITS^{[1][5]}.

The P-matrix of "side by side" combination of N circuits as shown in Fig. 3 has $(n_1 + \dots + n_N)$ input variables and $(m_1 + \dots + m_N)$ output variables and is given by

$$[P] = [P_1] \otimes [P_2] \otimes \dots \otimes [P_N]$$

where $\otimes$ denotes Kronecker product.

4.4 PARALLEL COMBINATION OF CIRCUITS^[5]. The P-matrix of parallel combination of N circuits as shown in Fig. 4 has n input variables and $(m_1 + \dots + m_N)$ output variables and is given by

$$[P] = [P_1] * [P_2] * \dots * [P_N]$$

where * denotes the row multiplication which is defined as

$$p_{i,j} = p\{j/i\} = \sum_{k=1}^{\pi} p_{i,j_k}$$

and $p_{i,j}$ is the (i,j) entry of $[P]$, p_{i,j_k} is an (i,j_k) entry of $[P_k]$,

$i = 0, 1, \dots, 2^n - 1$; $j_k = 0, \dots, 2^{m_k} - 1$; $k = 1, 2, \dots, N$;

$j = 0, 1, \dots, (2^{m_1 + \dots + m_N} - 1) = j_1 \times 2^{m_2 + \dots + m_N} + \dots + j_{N-1} \times 2^{m_N} + j_N$

4.5 P-MATRIX OF PROBABILISTIC SEQUENTIAL CIRCUIT^[5].

By considering the i-th probabilistic unit delay element as two logical elements in series. The first element is a hypothetical logical element A_i having the same probabilistic characteristic of the unit delay element and the second element is a perfect unit delay element D_{pi} . The modified sequential circuit, of Fig. 1, is shown in Fig. 5. Assume that the P-matrix of the i-th unit delay element is

$$[P_{di}] = \begin{bmatrix} \begin{matrix} s_i(k+1) & s_i(k) \\ 0 & 1 \end{matrix} & \begin{matrix} 0 \\ 1 \end{matrix} \\ \begin{matrix} 1-a_i \\ b_i \end{matrix} & \begin{matrix} a_i \\ 1-b_i \end{matrix} \end{bmatrix}$$

Thus

$$[P_{Ai}] = \begin{matrix} \begin{matrix} s_i'(k+1) \\ s_i(k+1) \end{matrix} & \begin{matrix} 0 & 1 \end{matrix} \\ \begin{matrix} 0 \\ 1 \end{matrix} & \begin{bmatrix} 1-a_i & a_i \\ b_i & 1-b_i \end{bmatrix} \end{matrix} \quad i = 1, 2, \dots, r.$$

Since $S(k+1)$ and $S'(k+1)$ are one to one correspondences, the $S'(k+1)$ can be relabelled as $S(k+1)$.

We can consider that p. s. c. consists of three parts in series, the first part is the combinatorial circuit, the second part are the output lines and the hypothetical logical elements, and the third part are the perfect unit delay elements. Thus P-matrix of the p. s. c. is

$$[P'] = [P_c] \cdot \{ [P_{A1}] \otimes [P_{A2}] \otimes \dots \otimes [P_{Ar}] \otimes \underbrace{[I] \otimes \dots \otimes [I]}_m \}$$

$$\text{or} \quad = [P_c] \cdot \{ \underbrace{[I] \otimes \dots \otimes [I]}_m \otimes [P_{A1}] \otimes \dots \otimes [P_{Ar}] \}$$

where $[I]$ is 2×2 matrix.

5. MARKOV MODEL FOR BERNOULLI INPUT WITHOUT NOISE DISTURBANCE

From equations (2), (3) and the explanation that follows these equations, the following statements are true:

1. The probability of the correct next state r -tuple depends on the present input n -tuple and present state r -tuple only.
2. The error in the actual logical operation of the p. s. c. causes the output m -tuple and/or the next state r -tuple to differ from the answer provided by the equations (2) and (3) for a given present state r -tuple and present input n -tuple.

We are now in a position to derive a modified P-matrix, $[P'']$, of the p. s. c. . The P-matrix of the combinatorial circuit portion of the p. s. c. , $[P_c]$, can be formed systematically by applying the "*", ".", and "x" operators to the component P-matrices. Let e be the state into which are lumped all the erroneous state transitions and erroneous output m -tuples for each present state r -tuple and input n -tuple of the p. s. c. . Since we are going to find the mean number of times that the p. s. c. stays in its correct states and correct output m -tuple before committing an error, we can make the state e an absorbing state, $p_{e,e}=1$, $p_{e,j}=0$ and $p_{i,e}=r_i$; $j=0, \dots, 2^{m+r}-1$; $i=0, \dots, 2^{n+r}-1$. The r_i includes all the erroneous state transitions and erroneous output m -tuples. From equations (2) and (3) we can find the correct next state r -tuple and output m -tuple for a given $(X(k), S(k))$. Since for each $(X(k), S(k))$ there is a probability of producing one and only one correct next state r -tuple and correct output m -tuple and the present output m -tuple does not affect the operations of the p. s. c. at time $t=k+1, k+2, \dots$, we can lump all the columns with headings $S(k+1)=i_r$ and $Y(k)=j$, $j=0, \dots, 2^m-1$, into a column with heading $X(k+1)=i$, $i=0, \dots, 2^r-1$. The resulting matrix is $[P'']$ which has a dimension of $(2^{n+r}+1) \times (2^r+1)$. Since we are dealing with Bernoulli input, the $[P'']$ is divided into 2^n submatrices, $\bar{P}(0), \bar{P}(1), \dots, \bar{P}(2^n-1)$ with respect to $X(k)=0, 1, \dots, 2^n-1$. Each of these submatrices,

$P\{e, S(k+1)/e, S(k)/X=i\} = \bar{P}(i); i = 0, \dots, 2^n-1$, is a $(2^r+1) \times (2^r+1)$ matrix. Since the input n -tuples are mutually independent, on the basis of total probability formula [6] the total transition matrix of the p. s. c. is

$$[P_T] = \sum_{i=0}^{2^n-1} p_i \cdot \bar{P}(i) \quad (4)$$

It is a stochastic matrix and the state transitions of the p. s. c. are described by a simple Markov chain with transition matrix $[P_T]$ [7].

Let $[P'_T]$ be the transition matrix of correct operations and $[r]$ be the column matrix of erroneous operations. Equation (4) becomes

$$[P_T] = S(k) \left\{ \begin{array}{c} e \\ 0 \dots 0 \\ \vdots \\ 1 \dots 1 \end{array} \right\} \left[\begin{array}{c} 1 \\ r \end{array} \right] \left[\begin{array}{c} \overbrace{0 \dots 0 \dots 1 \dots 1}^{S(k+1)} \\ 0 \dots 0 \\ \vdots \\ P'_T \end{array} \right] \quad (5)$$

6. MARKOV MODEL FOR BERNOULLI INPUT WITH BINARY NOISE DISTURBANCE

The Markov model for Bernoulli input with binary noise disturbance is quite similar to that of the case of perfect Bernoulli input. The only difference is that there are errors in the input n -tuples for the Bernoulli input with binary noise disturbance, and there are no errors in the input n -tuples for the Bernoulli input without binary noise disturbance. Let $p'_{i,j}$ ($i, j=0, 1, \dots, 2^n-1$) be the actual probability of input n -tuple j to the p. s. c. when the intended input n -tuple was i . The probability of correct input n -tuple of the p. s. c. is $p'_{i,i}$. Usually $p'_{i,i}$ will be nearly equal to p_i and $p'_{i,j}$ ($j \neq i$) will be very small.

Let $[\gamma]_i$ be the column matrix of the probabilities of erroneous operations of the p. s. c. when the input n -tuple is i , $[P_p]_i$ be the transition matrix of correct transitions when the input n -tuple is i , $i=0, 1, \dots, 2^n-1$. The matrix of correct transitions is the product of the probability of correct input and the probability of correct operation of the p. s. c., thus

$$[P'_T] = \sum_{i=0}^{2^n-1} p'_{i,i} \cdot [P_p]_i \quad (6)$$

The erroneous transitions are due to the incorrect input n -tuples while the operations of the p. s. c. are correct, or the erroneous operations of the p. s. c. when the input n -tuples are correct, thus

$$[r] = \sum_{i=0}^{2^n-1} \sum_{\substack{j=0 \\ j \neq i}}^{2^n-1} p'_{i,j} \cdot [C] + \sum_{i=0}^{2^n-1} p'_{i,i} \cdot [\gamma]_i \quad (7)$$

where $[C]$ is a column matrix with all entries 1 and with 2^r rows.

The total transition matrix $[P_T]$ is of the form of equation (5). For the same reason as in the case of Bernoulli input without binary noise disturbance, $[P_T]$ is a simple Markov chain.

7. A METHOD FOR FINDING THE RELIABILITY OF TIME-DISCRETE SYSTEM BY USING Z-TRANSFORM AND FLOW-GRAPH TECHNIQUES

Since the p. s. c. with Bernoulli input can be represented by a simple Markov chain, a set of difference equations is found.

$$Q(k+1) = Q(k) \cdot [P_T^1] \quad (8)$$

where $Q(k) = (q_0(k) \dots q_{2^r-1}(k))$ row vector of successful states, $[P_T^1]$ is the transition matrix of the system between successful states, and $q_i(k)$ is the probability that the system at the k -th time step is in the state i , $i=0,1,\dots,2^r-1$.

Since the flow-graph technique is applicable only to the solution of linear algebraic equations, in order to apply the flow-graph technique to find the reliability of a time-discrete system, the set of difference equations must be first transformed into a set of linear algebraic equations. This can be accomplished by z -transform. Taking the z -transform^[8] of equation (8) yields

$$Q(z) = z \cdot Q(z) \cdot [P_T^1] + Q(0) \quad (9)$$

where $Q(0) = (q_0(0) \dots q_{2^r-1}(0))$, $Q(z) = (q_0(z) \dots q_{2^r-1}(z))$, and $q_i(z) = \sum_{k=0}^{\infty} q_i(k)z^k = \sum_{k=0}^{\infty} (q_i(k))z^k$; $i = 0, 1, \dots, 2^r - 1$.

Let $p(k)$ be the probability that the system will fail at the k -th time step, then

$$R(m) = \sum_{k=m+1}^{\infty} p(k) = \text{reliability of the system in the time interval 0-th to m-th step}$$

$$= \sum_{i=0}^{2^r-1} q_i(m)$$

and

$$p(k) = \sum_{m=k}^{\infty} p(m) - \sum_{m=k+1}^{\infty} p(m) = R(k-1) - R(k)$$

Let

$$R(z) = \sum_{k=0}^{\infty} R(k)z^k = \sum_{k=0}^{\infty} \sum_{i=0}^{2^r-1} q_i(k)z^k = \sum_{i=0}^{2^r-1} q_i(z)$$

The mean-time-to-first-system-failure, T_m , is defined by

$$T_m = \sum_{k=0}^{\infty} k \cdot p(k) = \sum_{k=0}^{\infty} k \cdot (R(k-1) - R(k))$$

$$= \frac{d}{dz} \left[\sum_{k=0}^{\infty} R(k-1)z^k - \sum_{k=0}^{\infty} R(k)z^k \right] \Big|_{z=1} = R(z) \Big|_{z=1} = \sum_{i=0}^{2^r-1} q_i(z) \Big|_{z=1} \quad (10)$$

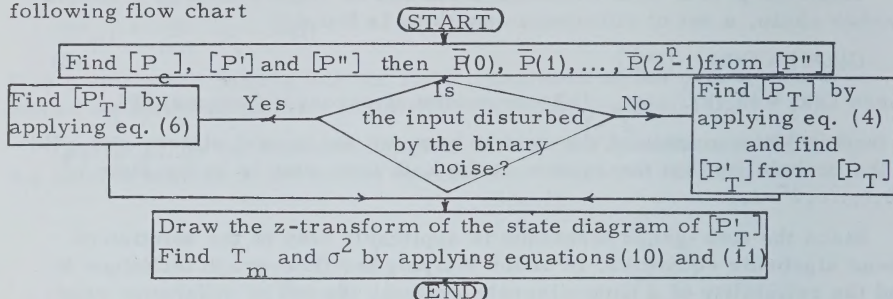
and the variance, σ^2 , is

$$\sigma^2 = 2 \frac{dR(z)}{dz} \Big|_{z=1} + R(z) \Big|_{z=1} - T_m^2 \quad (11)$$

Apply the flow-graph technique to equation (9) with $q_i(0)=1$ and $q_j(0)=0$, $i,j=0,1,\dots,2^r-1$, $j \neq i$, at a time then $q_k(z)/(q_i(0)=1)$ can be found, T_m 's and σ^2 's follow. The symbol k signifies the number of all successful states and $q_k(z)/(q_i(0)=1)$ is the z -transform of the probability of the system being in state k if it starts at state i .

8. PROCEDURE FOR FINDING THE RELIABILITY OF PROBABILISTIC SEQUENTIAL CIRCUIT WITH BERNOULLI INPUT

The algorithm for finding the reliability of p. s. c. is shown in the following flow chart



9. AN EXAMPLE

Let us consider a binary counter as shown in Fig. 6. The component P-matrices of AND, OR, NOT and Unit delay elements are given as $[P_{\cap}]$, $[P_{\cup}]$, $[P_{-}]$ and $[P_d]$ respectively,

$$\begin{aligned}
 [P_{\cap}] &= \begin{array}{c|cc} \text{Output} & 0 & 1 \\ \hline \text{Input} & & \\ 0 & 0 & 0 \\ 0 & 1 & 1-a_0 \\ 1 & 0 & a_0 \\ 1 & 1 & 1-a_1 \\ & & a_1 \\ & & 1-a_2 \\ & & a_2 \\ & & 1-a_3 \\ & & a_3 \end{array}; [P_{\cup}] = \begin{array}{c|cc} \text{Output} & 0 & 1 \\ \hline \text{Input} & & \\ 0 & 0 & \beta_0 \\ 0 & 1 & 1-\beta_0 \\ 1 & 0 & \beta_1 \\ 1 & 1 & 1-\beta_1 \\ & & \beta_2 \\ & & 1-\beta_2 \\ & & \beta_3 \\ & & 1-\beta_3 \end{array} \\
 [P_{-}] &= \begin{array}{c|cc} \text{Output} & 0 & 1 \\ \hline \text{Input} & & \\ 0 & \gamma_0 & 1-\gamma_0 \\ 1 & 1-\gamma_0 & \gamma_0 \end{array}; [P_d] = \begin{array}{c|cc} S(k) & 0 & 1 \\ \hline S(k+1) & & \\ 0 & 1-a & a \\ 1 & b & 1-b \end{array}
 \end{aligned}$$

$$n=m=r=1$$

Assume that a_i 's = β_i 's = γ_i 's = b ($i=0,1,2,3$) are sufficiently small so that the second and higher order terms can be neglected.

The P-matrix of the circuit between $(X(k), S(k))$ and $S(k+1)$ (Fig. 6) is

$$[P] = \{([P_{-}] \otimes [I]) \cdot [P_{\cap}]\} * \{([I] \otimes [P_{-}]) \cdot [P_{\cup}]\} \cdot [P_d]$$

and the P-matrix of the combinatorial circuit portion (Fig. 6) is

$$[P_c] = [P] * [P_{\cap}]$$

The P-matrix of the sequential circuit is

$$[P'] = [P_c] \cdot ([P_d] \otimes [I])$$

and

$$[P'] = X(k), S(k) \quad \begin{matrix} & & \underbrace{S(k+1)} \\ & & \begin{matrix} 0 & 1 \end{matrix} \\ \begin{matrix} e \\ e \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 5a & 1-5a & 0 \\ 5a & 0 & 1-5a \\ 5a & 0 & 1-5a \\ 7a & 1-7a & 0 \end{bmatrix} \end{matrix}$$

Assume that $p_0 = p_1 = \frac{1}{2}$ without binary noise disturbance, then the total transition matrix is

$$[P_T] = S(k) \quad \begin{matrix} & & \underbrace{S(k+1)} \\ & & \begin{matrix} 0 & 1 \end{matrix} \\ \begin{matrix} e \\ 0 \\ 1 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 5a & \frac{1}{2}(1-5a) & \frac{1}{2}(1-5a) \\ 6a & \frac{1}{2}(1-7a) & \frac{1}{2}(1-5a) \end{bmatrix} \end{matrix}$$

The z-transform of the state diagram of $[P_T']$ is shown in Fig. 7. Now apply the flow-graph technique to find T_{m1}' 's and σ_2^2 's.

$$\Delta = \frac{1}{2}(2 - 2(1-5a)z + az^2)$$

Starting at "0" state

$$q_0(z) = \frac{1}{\Delta} \left(1 - \frac{1}{2}(1-5a)z\right); \quad q_1(z) = \frac{1}{2\Delta} (1-5a)z$$

thus

$$T_{m0} = (q_0(z) + q_1(z)) \Big|_{z=1} = 2/11a$$

and

$$\sigma_0^2 = (4-26a)/121a^2; \quad 1 \gg a$$

Similarly for the circuit starts at "1" state

$$T_{m1} = 2(1-a)/11a; \quad \sigma_1^2 = (4-26a)/121a^2$$

In the case of Bernoulli input with binary noise disturbance and under the assumption that

$$p'_{0,0} = \frac{1}{2}(1-a) = p'_{1,1}; \quad p'_{0,1} = p'_{1,0} = \frac{1}{2}a,$$

then

$$[P_T'] = \begin{matrix} & \begin{matrix} \begin{matrix} S(k+1) \\ S(k) \end{matrix} \\ 0 \end{matrix} & \begin{bmatrix} 1 & 0 \\ \frac{1}{2}(1-6a) & \frac{1}{2}(1-6a) \\ \frac{1}{2}(1-8a) & \frac{1}{2}(1-6a) \end{bmatrix} \\ 1 & \end{matrix}$$

and

$$T_{m0} = 2/13a; \quad \sigma_0^2 = (4-30a)/169a^2$$

$$T_{m1} = 2(1-a)/13a; \quad \sigma_1^2 = (4-30a)/169a^2$$

10. CONCLUSION

A systematic way for finding the Markov chain model of p. c. s. with Bernoulli input with or without binary noise disturbance while the component P-matrices are known is presented. Also a mathematical formulation has been established for finding the reliability of time-discrete system with constant failure rate. The flow-graph technique is found very useful in finding the reliability of p. c. s. i. e. the mean-time-to-first-system-failure, T_m , and variance, σ^2 .

11. REFERENCES

1. K. Udagawa and Y. Inagaki, "Probabilistic studies of logical circuits and the synthesis of reliable circuits using probabilistic logical matrix", Journal of the Institute of Electrical Communication Engineers in Japan, Vol. 46, No. 11, pp. 195-206, Nov. 1963.
2. V. I. Levin, "Probabilistic analysis of combination systems and their reliability", Engineering Cybernetics, No. 6, pp. 78-84, Nov.-Dec. 1964.
3. K. Udagawa and Y. Inagaki, "Probabilistic studies of sequential circuits and estimation of their reliability using Markov chain", Journal of the Institute of Electrical Communication Engineers in Japan, Vol. 47, No. 4, pp. 276-286, April 1964.
4. G. H. Mealy, "Method for synthesizing sequential circuit", Bell System Technical Journal, Vol. 34, pp. 1045-1079, 1955.
5. G. S. Glinski and Who-Kee Chung, "A Markov model for reliability of probabilistic sequential circuit with Bernoulli inputs", T.R. No. 68-4, Department of Electrical Engineering, The University of Ottawa, Ottawa, Canada, May 1968.
6. A. Papoulis, "Probability, random variables, and stochastic processes", McGraw-Hill Book Co., New York, 1965.
7. W. Feller, "An introduction to probability theory and its applications", Vol. I, John Wiley and Sons, Inc., New York, 1957.
8. R. A. Howard, "Dynamic programming and Markov processes", Technology Press of MIT and John Wiley and Sons, Inc., New York, 1960.

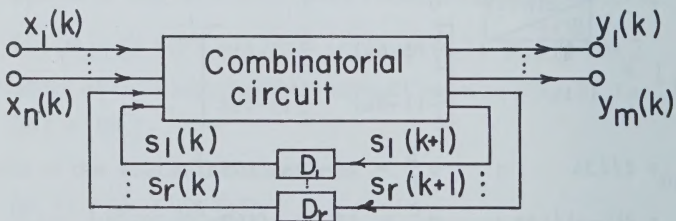


Fig. 1. A sequential circuit

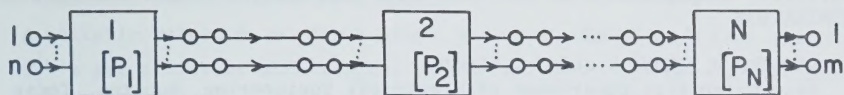


Fig. 2. Series combination of N circuits

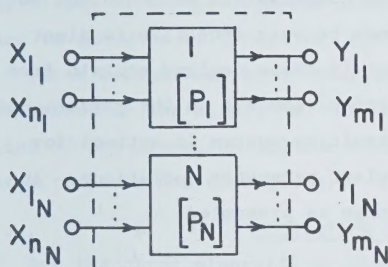


Fig. 3. "Side by Side" combination of N circuits

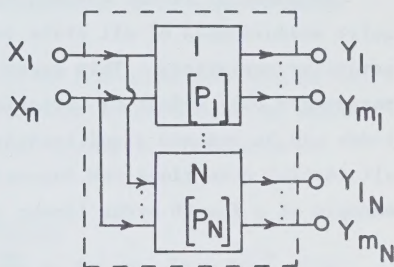


Fig. 4. Parallel combination of N circuits

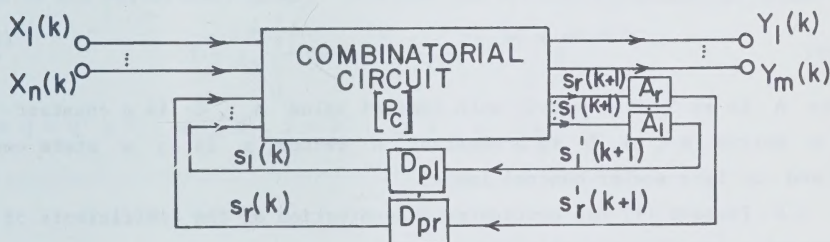


Fig. 5. Modified sequential circuit of Fig. 1

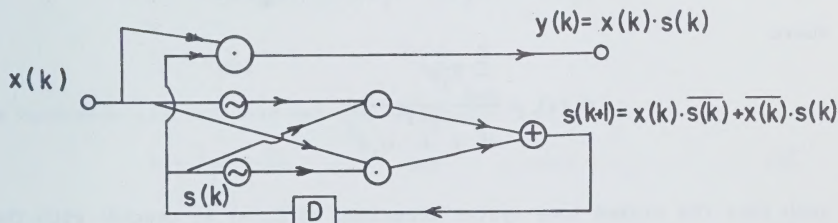


Fig. 6. Binary counter

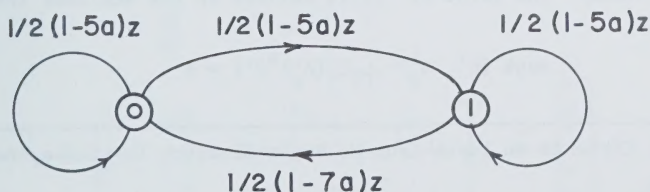


Fig. 7. The z -transform of the state diagram of $[P_T^I]$

SENSITIVITY CONSIDERATION OF MULTIPLE INPUT COMPENSATOR DESIGN FOR DYNAMIC OPTIMIZATION

VIDOJKO V. CIRIC * and J.V. LEEDS

Rice University, Department of Electrical Engineering, Houston, Texas

SUMMARY

Response of linear systems with optimum compensators which do not require measurements of all state variables may be very sensitive to plant parameter variations. This paper shows that if these designs contain free parameters, the effect of variations of plant parameters on the performance index can be reduced significantly. The resulting system is optimal for all initial conditions and insensitive to plant parameter variations. An example of a fourth order linear system design is presented.

INTRODUCTION

Consider a completely controllable and completely observable linear plant

$$\begin{aligned}\dot{x} &= Ax + bu \\ y &= Cx\end{aligned}\tag{1}$$

where A is an $n \times n$ matrix with nominal value A_0 , C is a constant $m \times n$ matrix ($m < n$), b is a constant n vector, x is an n state vector and u is a scalar control input.

J.B. Pearson [1] has considered determination of the coefficients of a compensator transfer function matrix

$$G_c(s) = [G_1(s) \ G_2(s) \ \dots \ G_m(s)]\tag{2}$$

where

$$G_j(s) = \frac{\sum_{i=0}^p \beta_i^j s^i}{s^p + \sum_{i=0}^{p-1} \alpha_i s^i}$$

such that the closed loop system comprised of $G_c(s)$ in cascade with the plant has a given set $\Lambda = \{\lambda_1, \dots, \lambda_{n+p}\}$ of poles and minimizes a quadratic performance index. The value of p is defined as the smallest integer such that

$$\text{rank } [C', A_0' C', \dots, (A_0')^p C'] = n$$

* Vidojko V. Ciric is on leave from M. Pupin Research Institute, Belgrade, Yugoslavia.

Prime indicates transposition of a matrix.

A suitable value of p for a given m can be found such that some of the β_i^j 's are arbitrary with respect to the optimality of the nominal plant. These arbitrary β_i^j 's can be used to minimize the sensitivity of the performance index with respect to plant parameter variations. This paper presents an algorithm for finding values of the β_i^j 's which minimize the sensitivity of the performance index. The sensitivity of the performance index is found to be linear in β_i^j 's.

First a brief review of Pearson's results is presented followed by the development of the algorithm. The computer results illustrating the application of the algorithm and the effect of plant parameter variations on system response are also given.

DERIVATION OF COMPENSATOR

For the given plant (1) of the n^{th} order, an optimal $(n+p)^{\text{th}}$ order system results by weighing the derivatives of the plant input in the integrand of the performance index. Consider the following performance index

$$I = \int_0^\infty \left\{ x' Q x + \sum_{i=0}^p \gamma_i [u^{(i)}]^2 \right\} dt \quad (3)$$

where $Q = Q' \geq 0$, $\gamma_i \geq 0$, $i = 0, \dots, p-1$, $\gamma_p \equiv \gamma > 0$.

Let

$$\begin{aligned} u &= u_1 \\ \dot{u} &= u_2 \\ &\vdots \\ \dot{u}_p &= u_{p+1} \equiv U. \end{aligned} \quad (4)$$

Then the equations (1), (3) and (4) can be written as:

$$\dot{z} = \bar{A}_0 z + \bar{b} U, \quad z(0) = z_0 \quad (5)$$

$$I = \int_0^\infty (z' \bar{Q} z + \gamma U^2) dt \quad (6)$$

where

$$z = \begin{bmatrix} x \\ u \end{bmatrix}; \quad u = \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix}$$

$$\bar{A}_0 = \begin{bmatrix} A & b0 \\ 0 & B \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 & & & \\ & \ddots & \ddots & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 & 0 \end{bmatrix} \quad \bar{b} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

$$\bar{Q} = \begin{bmatrix} Q & 0 \\ 0 & \Gamma \end{bmatrix} \quad \Gamma = \begin{bmatrix} \gamma_0 & 0 \\ 0 & \gamma_{p-1} \end{bmatrix}.$$

For the system (5) and (6) we can obtain [2]

$$U = -\bar{k}' \cdot z = - \sum_{i=1}^m k_i x_i - \sum_{i=1}^p k_{n+i} u_i \quad (8)$$

where the vector $\bar{k} = [k_1, \dots, k_n, k_{n+1}, \dots, k_{n+p}]'$ is obtained from the equation

$$\frac{1}{\gamma} P \cdot \bar{b} = \bar{k} \quad (9)$$

Matrix P is the steady-state solution of the matrix Riccati equation

$$- \dot{P} = P \bar{A}_0 + \bar{A}_0' P + \bar{Q} - \frac{1}{\gamma} P \bar{b} \bar{b}' P \quad (10)$$

with $P(0) = 0$.

On the other hand by using Eq. (2) the compensated system can be written in the same form as system (5) with the last equation of the form

$$U = - \sum_{i=0}^{p-1} \alpha_i u_{i+1} - \sum_{i=0}^p \sum_{j=1}^m \beta_i^j y_j^{(i)} \quad (11)$$

where $y_j^{(i)} = c_j' x^{(i)}$ and c_j' is j^{th} row vector of matrix C . From (8) and (11) it follows that:

$$\sum_{i=0}^p \sum_{j=1}^m \beta_i^j c_j' A_0^i x = \sum_{i=1}^n k_i x_i \quad (12)$$

$$\sum_{i=1}^p \alpha_{i-1} u_i + \sum_{i=1}^p \sum_{j=1}^m \beta_i^j \sum_{k=1}^i A_0^{i-k} b u_k = \sum_{i=1}^p k_{i+n} u_i \quad (13)$$

Equation (12) must hold for all x , which implies

$$[C', A_0' C', \dots, (A_0')^p C'] \begin{bmatrix} \beta_0^1 \\ \beta_0^2 \\ \vdots \\ \beta_p^1 \\ \vdots \\ \beta_p^m \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} \quad (14)$$

Once β_i^j 's are found, α_i can be uniquely determined from equation (13).

DERIVATION OF ALGORITHM

The results presented below are preliminary. Full details will be reported in reference [10] and others to be published.

The solution of equations (13) and (14) do not always result in unique β_i^j 's. Equation (14) represents a system of n equation in $m(p+1)$ unknowns. If $m(p+1) > n$, $m(p+1)-n = k$ of the β_i^j 's can be assigned arbitrarily and the system is still optimal regardless of their values. However, the system can be very sensitive to plant parameter variations if the arbitrary (free) β_i^j 's are not properly chosen. If the system (14) does not have arbitrary β_i^j 's, one can obtain them by increasing the value of p .

We now find values for the free β_i^j 's which minimize the sensitivity of the system performance index with respect to plant parameter variations, i.e., we find the vector β , whose components are the free β_i^j 's, that minimizes

$$J = \left\| \frac{\partial I}{\partial a} \right\|_2^2 = \left(\frac{\partial I}{\partial a} \right)_{a_0}' \left(\frac{\partial I}{\partial a} \right)_{a_0} \quad (15)$$

Let us assume that ℓ plant parameters can vary and that there are k free β_i^j 's to be determined. The plant parameters, which are the elements of the matrix A , are represented by the vector a .

Consider now a more general problem

$$\dot{z} = f(z, U(z, \beta), a) \quad , \quad z(0) = z_0 \quad (16)$$

$$I(a, \beta) = \int_0^\infty G(z, U(z, \beta)) dt \quad (17)$$

and assume that all needed partial derivatives exist.

The first variation of the performance index is given by

$$\Delta I = \left(\frac{\partial I}{\partial a} \right)_{a_0}' \cdot \Delta a \quad (18)$$

where $\Delta a = a - a_0$ is an ℓ column vector and

$$\left(\frac{\partial I}{\partial a} \right)' = \int_0^\infty \left(\frac{\partial G}{\partial z} + \frac{\partial G}{\partial U} \cdot \frac{\partial U}{\partial z} \right) \frac{\partial z}{\partial a} dt = \int_0^\infty F \frac{\partial z}{\partial a} dt \quad (19)$$

All partial derivatives are evaluated at $a = a_0$.

The sensitivity matrix is determined from the variation equation obtained from (16) at the point (z, U, a_0) :

$$\frac{d}{dt} \left(\frac{\partial z}{\partial a} \right) = \varphi_1(t) \frac{\partial z}{\partial a} + \varphi_2(t), \quad \frac{\partial z}{\partial a} \Big|_{t=0} = 0 \quad (20)$$

with

$$\varphi_1(t) = \frac{\partial f}{\partial z} + \frac{\partial f}{\partial U} \cdot \frac{\partial U}{\partial z}, \quad \varphi_2(t) = \frac{\partial f}{\partial a}, \quad (21)$$

where $\frac{\partial f}{\partial U}$, $\frac{\partial f}{\partial z}$, and $\frac{\partial f}{\partial a}$ are evaluated along the nominal optimal trajectory.

For the system (5) and (6) the above quantities are

$$F = z' \cdot \bar{K} \quad \text{where} \quad \bar{K} = 2(\bar{Q} + \gamma \cdot \bar{k} \cdot \bar{k}') \quad (22)$$

$$\varphi_1 = \bar{A}_0 - \bar{b} \bar{k}' = \bar{A}_0 \quad \text{where} \quad \bar{A} - \bar{b} \bar{k}' = \bar{A} \quad (23)$$

$$\varphi_2(t) = \frac{\partial}{\partial a} (\bar{A}) z(a, t) \Big|_{a=a_0} = \frac{\partial}{\partial a} (\bar{A}) \Big|_{a=a_0} \cdot z(a_0, t) \quad (24)$$

Therefore

$$\left(\frac{\partial I}{\partial a} \right)' (a_0, \beta) = \int_0^\infty z' (a_0, t) \bar{K} \frac{\partial z}{\partial a} (a_0, \beta, t) dt \quad (25)$$

Note, that the system has the following properties:

$$z(a_0, \beta, t) = z(a_0, t) \quad (26)$$

$$I(a_0, \beta) = I(a_0).$$

Lemma 1. Vector $\left(\frac{\partial I}{\partial a} \right)'$ is linear in β and can be written as

$$\left(\frac{\partial I}{\partial a} \right)' = (d + D\beta)' = d' + \beta' D' \quad (27)$$

where d is ℓ column vector and D is $\ell \times k$ matrix:

$$d' = \int_0^\infty \int_0^\infty z' (a_0, t) \bar{K} e^{\bar{A}_0(t-\tau)} \cdot \left(\frac{\partial \bar{A}}{\partial a} \right)_{\beta=0} \cdot z(a_0, \tau) d\tau dt \quad (28)$$

$$D' = \int_0^\infty \int_0^\infty [z' (a_0, t) \cdot \bar{K} \cdot e^{\bar{A}_0(t-\tau)} \cdot \bar{b}] (-\bar{K}' \cdot \frac{\partial \bar{A}}{\partial a})_{a_0} \quad (29)$$

$$\cdot z(a_0, \tau) d\tau dt.$$

Proof. The solution of equation (20) is given by

$$\frac{\partial z}{\partial a} (a_0, \beta, t) = \int_0^\infty e^{\bar{A}_0(t-\tau)} \cdot \frac{\partial}{\partial a} \bar{A}(a, \beta)_{a_0} \cdot z(a_0, \tau) d\tau \quad (30)$$

where $M \equiv \frac{\partial}{\partial a} \bar{A}$ is an $(n+p) \times (n+p) \times \ell$ tensor whose ijp element is

$$m_{ijp} = \frac{\partial}{\partial a_p} (\bar{A}_{ij}) \quad (31)$$

If

$$\frac{\partial}{\partial a} \bar{A} = \left(\frac{\partial}{\partial a} \bar{A} \right)^0 + \left(\frac{\partial}{\partial a} \bar{A} \right)_\beta \quad (32)$$

then

$$\frac{\partial z}{\partial a} (a_0, \beta, t) = \left(\frac{\partial z}{\partial a} \right)^0 + \left(\frac{\partial z}{\partial a} \right)_\beta \quad (33)$$

where the first term on the right hand side of each above equation does not depend on β .

In order to show (32), we need the result of lemma 2 proved in reference [10].

Lemma 2. For $p=1$, the last row of the matrix

$$\bar{A} = \bar{A} - \bar{b} \cdot \bar{k}' \quad (34)$$

denoted by the row vector $\bar{k}'$ has the following general form

$$\bar{k}' = \bar{k}' + \bar{\gamma}' (\bar{A} - \bar{A}_0) + \beta' \bar{k}' (\bar{A} - \bar{A}_0), \quad (35)$$

where $\bar{\gamma}$ is an $(n+p)$ constant vector, $\bar{k}$ is an $(n+p) \times k$ constant matrix, $\bar{A}_0$ denotes nominal value of the matrix $\bar{A}$.

Both, $\bar{\gamma}$ and $\bar{k}$ depend only on $\bar{A}_0$ and $\bar{k}$. Therefore, from Eq. (32), (34) and (35), it follows that

$$\left(\frac{\partial}{\partial a} \bar{A} \right)^0 = [I - \bar{b} \cdot \bar{\gamma}'] \frac{\partial \bar{A}}{\partial a} = \left(\frac{\partial \bar{A}}{\partial a} \right)_{\beta=0} \quad (36)$$

$$\left(\frac{\partial}{\partial a} \bar{A} \right)_\beta = -\bar{b} \cdot \beta' \cdot \bar{k}' \frac{\partial \bar{A}}{\partial a} \quad (37)$$

and by substituting those equations in (32), (30) and (28), we obtain (27).

This completes the proof.

For later use, we shall call

$$z(a_0, t) = e^{\bar{A}_0 t} \cdot z(0) = \varphi(t) \cdot z(0) \quad (38)$$

$$S = -\bar{k}' \cdot \left(\frac{\partial \bar{A}}{\partial a} \right)_{a_0} \quad (39)$$

$$g'(\tau) = \int_0^\infty [\varphi(t) \cdot z(0)]' \cdot \bar{k} \cdot \varphi(t - \tau) dt \quad (40)$$

Then equations (28) and (29) become

$$d' = \int_0^\infty g'(\tau) \left(\frac{\partial \bar{A}}{\partial a} \right)_{\beta=0} \cdot \varphi(\tau) \cdot z(0) d\tau \quad (41)$$

$$D' = \int_0^\infty [g'(\tau) \cdot \bar{b}] \cdot S \cdot \varphi(\tau) \cdot z(0) d\tau \quad (42)$$

Lemma 3. The expressions (41) and (42) can be written in the following closed form:

$$d' = v' T v \quad (43)$$

$$D' = S P \bar{M} v \quad (44)$$

where P is similarity matrix which diagonalizes the matrix $\bar{A}_0$, (*)
 $\Lambda = \{\lambda_i\} = \text{diag} (\bar{A}_0)$; T is an $(n+p) \times (n+p) \times \ell$ order tensor whose $(ij\rho)$ element is

$$t_{ij\rho} = - \frac{r_{ij\rho}}{\lambda_i + \lambda_j} \quad (45)$$

and $R = \{r_{ij\rho}\} = E P^{-1} \left(\frac{\partial \bar{A}}{\partial a} \right)_{\beta=0} P$. (46)

The (ij) element of the $(n+p) \times (n+p)$ matrix E is

$$e_{ij} = - \frac{n_{ij}}{\lambda_i + \lambda_j} \quad (47)$$

where n_{ij} is an element of the $(n+p) \times (n+p)$ matrix N ,

$$N = P' \cdot \bar{K} \cdot P \quad (48)$$

The (ii) element of the $(n+p) \times (n+p)$ diagonal matrix $\bar{M}$ is

$$m_{ii} = \sum_{s=1}^{n+p} - \frac{v_s \cdot w_s}{\lambda_i + \lambda_s} \quad (49)$$

where v_s and w_s are the s^{th} components of the vectors

$$v = P^{-1} \cdot z(o) \quad , \quad w = E \cdot P^{-1} \cdot \bar{b} \quad (50)$$

respectively.

The proof of lemma 3 is given in [10].

Due to (27), the expression for J , equation (15), is quadratic in β :

$$J = \beta' D' \cdot D \beta + 2 d' D \beta + d' \cdot d \quad (51)$$

Setting $\frac{\partial K}{\partial \beta} = 0$ implies

$$\beta^* = - (D' D)^{-1} \cdot D' d \quad (52)$$

Remark. For $k = \ell$ D is a square matrix and if D is nonsingular, equation (52) reduces to

$$\beta^* = - D^{-1} \cdot d \quad (53)$$

In that case, minimum value of norm J is equal to zero.

EXAMPLE -- FOURTH ORDER SYSTEM

We shall now work a fourth order example. Further theoretical results can be found in reference [10].

(*) We shall assume that matrix $\bar{A}_0$ has distinct eigenvalues.

Let the plant be

$$\begin{aligned}
 \dot{x}_1 &= A_{11}x_1 + x_2 \\
 \dot{x}_2 &= A_{22}x_2 + x_3 \\
 \dot{x}_3 &= -1.5x_3 + x_4 \\
 \dot{x}_4 &= -0.5x_4 + u \\
 y &= x_1 + x_2 + x_3 \\
 x(0) &= [1 \ 1 \ 1 \ 1]' ; a = [A_{11}A_{22}]' ; a_0 = [-2. \ -1.]'
 \end{aligned} \tag{54}$$

In this case

$$A = \begin{pmatrix} A_{11} & 1 & 0 & 0 \\ 0 & A_{22} & 1 & 0 \\ 0 & 0 & -1.5 & 1 \\ 0 & 0 & 0 & -0.5 \end{pmatrix} ; \quad b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} ; \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} .$$

The system is completely controllable since $\text{rank} [b, A_0 b, A_0^2 b, A_0^3 b] = 4$.

The system is also completely observable with observability index $p = 1$, because $\text{rank} [C', A_0' C'] = 4 = n$.

Since $n=4, m=3, p=1$, it follows that $k = m(p+1)-n = 2$. Therefore, the compensator has two free parameters. Consider the following index performance:

$$\begin{aligned}
 I &= \int_0^\infty (x_1^2 + x_2^2 + x_3^2 + x_4^2 + u^2 + \dot{u}^2) dt \\
 &= \int_0^\infty (x' Q x + u^2 + \dot{u}^2) dt
 \end{aligned} \tag{55}$$

where $\text{diag } Q = [1 \ 1 \ 1 \ 1]$, $\gamma_0 = 1$, $\gamma_1 \equiv \gamma = 1$.

The augmented system (not perturbed) is

$$\begin{aligned}
 \dot{z} &= \bar{A}_0 z + \bar{b} u, \quad z(0) = [1 \ 1 \ 1 \ 1 \ 1]' \\
 I &= \int_0^\infty (z' \bar{Q} z + \gamma u^2) dt
 \end{aligned} \tag{56}$$

where

$$\bar{A}_0 = \begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1.5 & 1 & 0 \\ 0 & 0 & 0 & -0.5 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} ; \quad \bar{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} ; \quad \bar{Q} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} .$$

From equation (14), if we choose $\beta = [\beta_1 \beta_2]' = [\beta_1^1 \beta_1^2]'$ to be the free vector, it follows that

$$\beta_1^3 = k_4 ; \beta_0^1 = k_1 + 2\beta_1 ; \beta_0^2 = k_2 - \beta_1 + \beta_2 ; \beta_0^3 = k_3 - \beta_2 + 1.5k_4 \quad (57)$$

From equation (13) we obtain $\alpha_0 = k_5$.

The structure of the compensator is given in figure 1, where

$$\begin{aligned} \xi_1 &= \beta_1^1 = \beta_1 ; \xi_2 = \beta_1^2 = \beta_2 ; \xi_3 = \beta_1^3 = k_4 \\ \tau_1 &= \beta_0^1 - \beta_1^2 \alpha_0 = k_1 + 2\beta_1 - \beta_1 k_5 \\ \tau_2 &= \beta_0^2 - \beta_1^2 \alpha_0 = k_2 - \beta_1 + \beta_2 - \beta_2 k_5 \\ \tau_3 &= \beta_0^3 - \beta_1^3 \alpha_0 = k_3 - \beta_2 + 1.5k_4 - k_4 k_5 \end{aligned} \quad (58)$$

From figure 1 we have

$$U = -\bar{k}' \cdot z = -\alpha_0 u - \xi_1 \dot{x}_1 - \xi_1 \alpha_0 x_1 - \xi_2 \dot{x}_2 - \xi_2 \alpha_0 x_2 - \xi_3 \dot{x}_3 - \xi_3 \alpha_0 x_3 \quad (59)$$

$$-\tau_1 x_1 - \tau_2 x_2 - \tau_3 x_3 = -[\bar{k}' + \bar{\gamma}'(\bar{A} - \bar{A}_0) + \beta' \bar{k}'(\bar{A} - \bar{A}_0)] z$$

Using Eq. (58) we obtain

$$\bar{\gamma} = [0 \ 0 \ 0 \ 0 \ 0]' ; \bar{k}' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

From a preliminary computer program, which handles problems of arbitrary dimension, the optimal value at the controller parameter vector is $\beta^* = [0.417, 1.588]'$. An independent check of the above result is in progress and will be reported soon.

Computer traces showing response variations to plant parameters and variations in β are shown in Figures 2 and 3. In tables 1 and 2 the values of the performance index are given for the above variations.

CONCLUSION

The proposed compensator design has the following features: The nominal system is optimal with respect to the given performance index for all initial conditions (arbitrary, but fixed). It is not necessary to measure all state variables. Furthermore, the system is insensitive to small parameter variations. Sensitivity performance index, from which β is obtained is quadratic in β . The expression for β^* is in the form (Equations 52, 43, 44) that only matrix manipulation are involved. Even though we have considered first order compensator ($p = 1$), the theory is easily extended for $p > 1$.

The computer results of fourth order linear system (Figures 2 and 3 and Tables 1 and 2) shows the importance of sensitivity consideration in compensator design. Namely, when design takes into account sensitivity consideration, the system performs well with respect to the performance index for a

wide range of plant parameter variations. However, when the system is designed only on the basis to be optimal for nominal plant, the performance index of the system can become very large, even infinite (see Figures 2 and 3), for very small plant parameter variations.

ACKNOWLEDGMENT

The authors wish to express their sincere appreciation to Professor J.B. Pearson for his invaluable advice, suggestions and discussions during the writing of this paper. Thanks are also due Bill Eads for his aid in computer programming. This work was supported in part by NSF-GU-1153.

REFERENCES

1. J.B. Pearson, "Compensator Design for Dynamic Optimization", To appear in International Journal on Automatic Control.
2. R.E. Kalman, "When is a Linear Control System Optimal?", Trans. ASME, J. Basic Eng., Ser. D, Vol. 86, pp 51-60, 1964.
3. D.G. Luenberger, "Observers for Multivariable Systems", IEEE Trans. A.C., Vol. AC-11, pp 190-197, April 1966.
4. W.M. Wonham, "Pole Assignment in Controllable Systems", IEEE Trans. A.C., Vol. AC-12, No. 6, pp 660-665, 1967.
5. D.M. Salmon and P. Kokotovic, "Multi-Point Design of Time-Varying Feedback Controllers", Proc. Fifth Allerton Conference on Circuit and System Theory, (University of Illinois, Urbana) October 4-6, 1967.
6. V. Ciric, S. De Julio, M. Vuskovic, "Sensitivity Analysis of the Control System for Rolling Plant on Analog Computer", Journal "L'Electrotecnica", Milano, Vol. LII, No. 12, 1965 (in Italian).
7. M. Vuskovic, V. Ciric, "Structural Rules for the Defermination at Sensitivity Functions of Nonlinear Nonstationary Systems", Proceedings of the Int. Symp. on Sensitivity Analysis, Dubrovnik, Sept. 1964.
8. M. Vuskovic, V. Ciric, "Application of the Sensitivity Method to the Analysis of the Blooming Drive Control System", II International Congress on Problems of Automation in Metallurgy, Budapest, 1965 (in Russian).
9. R.S. Varga, "Matrix Iterative Analysis", Prentice-Hall, 1965.
10. V. Ciric, J.V. Leeds, "Further results on Sensitivity Consideration of Multiple Input Compensator Design for Dynamic Optimization", To appear.

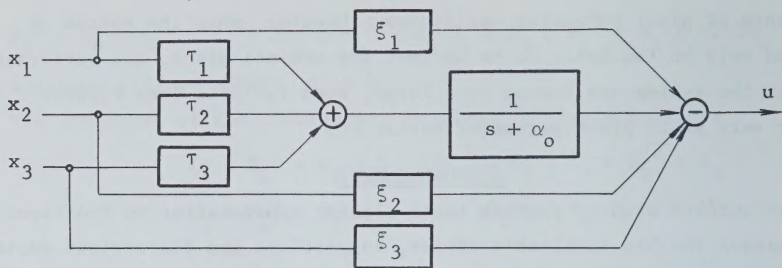


FIG. 1

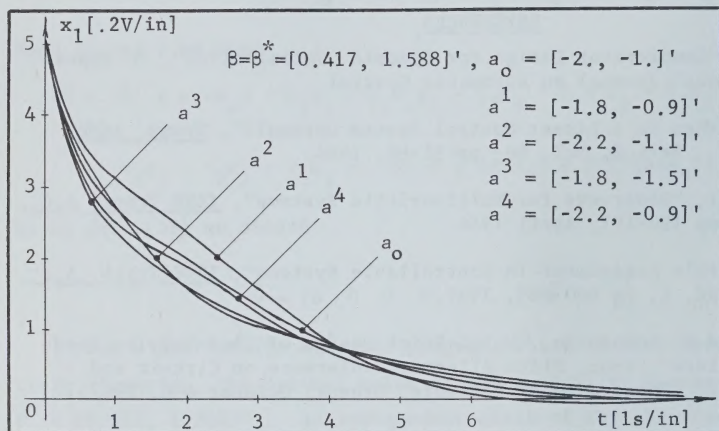


FIG. 2

TABLE 1

($\beta = \beta^*$)

a	I
a_0	7.20
a^1	7.45
a^2	7.49
a^3	8.68
a^4	7.32

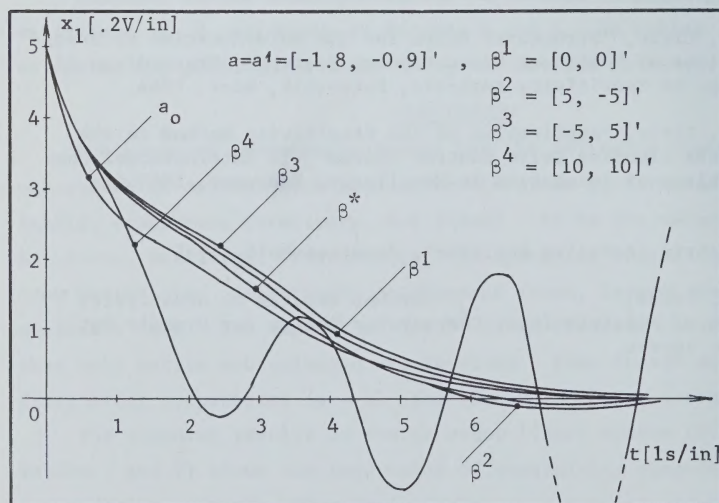


FIG. 3

TABLE 2

($a = a^1$)

β	I
β^1	7.52
β^2	7.60
β^3	7.71
β^4	∞
β^*	7.45

J. M. HOLTZMAN

Bell Telephone Laboratories, Incorporated, Whippany, New Jersey

ABSTRACT

A technique is presented to bound the error in using the equation of first variation to linearize about a nominal solution to a differential equation. The problem is treated as a special case of a problem of implicit functions.

I. INTRODUCTION

Sensitivity analysis is concerned with determining effects of parameter variations upon the solution of an equation, e.g., a differential equation. The most common approach used is that of assuming "sufficiently small" parameter variations and linearizing about a nominal solution. This technique, while of great practical importance, does suffer from the defect of not providing a measure of what is meant by "sufficiently small". That is, the accuracy of linearization is left unresolved. This paper presents an approach to determination of what is meant by "sufficiently small for the validity of linearization".

II. LINEARIZATION

We consider the following abstract problem which includes many sensitivity problems in systems theory. Let X , Y , and Z be Banach spaces. Let the product space $X \times Y$ be normed, say, $\|(x, y)\| = \|x\| + \|y\|$. We will not notationally distinguish between norms in different spaces, the usage being clear ($\|x\|$ for $x \in X$, $\|y\|$ for $y \in Y$, etc.). Suppose $P(x, y)$ maps a subset of $X \times Y$ into Z and it is known for some (x_0, y_0) that $P(x_0, y_0) = 0$ (the zero element of Z). The implicit function theorem ([1], pp. 686, 687) gives sufficient conditions for a positive answer to the following question: is there a function $F(x)$, such that $P(x, F(x)) = 0$ in a neighborhood of (x_0, y_0) ? It is also found that (under certain conditions; see [1], p. 689)*

$$F'(x_0) = -\left[P'_y(x_0, y_0)\right]^{-1} P'_x(x_0, y_0) \quad (\text{II-1})$$

* P'_y and P'_x are partial derivatives.

Thus, if we consider x to represent a parameter (it could be a scalar or much more general) and y the solution of the equation we are interested in, then a "first approximation" for changes around a nominal solution y_0 is

$$y \approx y_0 + F'(x_0)(x - x_0) \quad (\text{II-2})$$

The above is actually the basis for most sensitivity analysis that is done (see, e.g., [2], [3], [4]). As a concrete example, consider the differential equation

$$\dot{y}(t) = f(y(t), x), \quad y(0) = 0, \quad t \in [0, t_1] = T \quad (\text{II-3})$$

where x is a vector parameter. A sensitivity analysis is usually carried out with the equation of first variation

$$\begin{aligned} \delta \dot{y}(t) &= f_y(y_0(t), x_0) \delta y(t) + f_x(y_0(t), x_0) \delta x \\ \delta y(0) &= 0, \quad t \in T \end{aligned} \quad (\text{II-4})$$

where f_y and f_x are matrices of partial derivatives. The relationship between the equation of first variation and the abstract problem considered before is well known but for the purposes of this paper it is helpful to review that relationship.

III. RELATIONSHIP BETWEEN EQUATION OF FIRST VARIATION AND ABSTRACT PROBLEM

First, convenient Banach spaces must be defined. If the parameter x is a real-valued q -vector, let X be a q -dimensional real space with norm $\|x\| = \max_{i=1, \dots, q} |x_i|$. If y is a real-valued r -vector of time functions on $T = [0, t_1]$, let Y be a space of r -vectors of real valued continuously differentiable functions of time on T with initial condition $y(0) = 0$ (for a linear space) and with a convenient norm (an example is given in Section V). Z is a space of r -vectors of real-valued continuous functions of time on T with a convenient norm.

Now consider the operation P mapping a subset of $X \times Y$ into Z .

$$z = P(x, y) \quad (\text{III-1})$$

$$z(t) = \dot{y}(t) - f(y(t), x), \quad t \in T \quad (\text{III-2})$$

When $z = 0$, we clearly have a solution to the differential equation (II-3). It is easily shown that for most spaces and under mild conditions on f that the partial derivative operation $z = P'_{y(x_0, y_0)} y$ is defined by

$$z(t) = \dot{y}(t) - f_y(y_0(t), x_0)y(t) \quad t \in T \quad (\text{III-3})$$

(see, e.g., the discussion on pp. 681-684 of [1]). In order to use (II-1) and (II-2) the inverse of $P_y'(x_0, y_0)$ is needed. This is obtained from the solution of (III-3) for $y(t)$ in terms of $z(t)$. The solution of (III-3) is

$$y(t) = \int_0^t \Phi(t, s)z(s)ds \quad t \in T \quad (\text{III-4})$$

where $\Phi(t, s)$ is the transition matrix for

$$\dot{y}(t) = f_y(y_0(t), x_0)y(t) \quad (\text{III-5})$$

To use (II-1) and (II-2) the partial derivative operation $z = P_x'(x_0, y_0)x$ must still be found. This is given by (under mild conditions)

$$z(t) = -f_x(y_0(t), x_0)x \quad t \in T \quad (\text{III-6})$$

Putting (II-1), (II-2), (III-4), and (III-6) together, letting $y - y_0 = \delta y$, $x - x_0 = \delta x$ and using equality instead of approximate equality in (II-2) yields

$$\delta \dot{y}(t) = \int_0^t \Phi(t, s)f_x(y_0(t), x_0)\delta x ds \quad t \in T \quad (\text{III-7})$$

or

$$\begin{aligned} \delta \dot{y}(t) &= f_y(y_0(t), x_0)\delta y(t) + f_x(y_0(t), x_0)\delta x, \\ \delta y(0) &= 0, \quad t \in T, \end{aligned} \quad (\text{III-8})$$

the equation of first variation.

IV. ESTIMATING THE ACCURACY OF USING THE EQUATION OF FIRST VARIATION

In deriving (III-8) it was mentioned that the approximate equality of (II-2) was replaced by equality. We wish to estimate the accuracy of this approximation. In particular, a bound on $\|y - y_0 - F'(x_0)(x - x_0)\|$ is desired. If it is known that $\|x - x_0\| < \delta$ implies that $\|y - y_0\| \leq \varepsilon$ then

$$\|y-y_0 - P'(x_0)(x-x_0)\| \leq \|\Gamma\| \left[\sup_{\substack{\|x-x_0\| < \delta_1 \\ \|y-y_0\| < \varepsilon_1}} \|P'_x(x,y) - P'_x(x_0,y_0)\| \cdot \|x-x_0\| + \sup_{\substack{\|x-x_0\| < \delta_2 \\ \|y-y_0\| \leq \varepsilon_2}} \|P'_y(x,y) - P'_y(x_0,y_0)\| \cdot \|y-y_0\| \right] \quad (\text{IV-1})$$

where $\Gamma = [P'_y(x_0, y_0)]^{-1}$ and where $(\delta_1, \varepsilon_1, \delta_2, \varepsilon_2)$ is either $(\delta, \varepsilon, 0, \varepsilon)$ or $(\delta, 0, \delta, \varepsilon)$ whichever gives the lower bound (see [1], p. 686). The implicit function theorem ([1], pp. 686, 687) gives sufficient conditions for the existence of ε and δ but explicit determination of numerical ε and δ is left open. The question of explicit ε and δ is considered in [5] where the following is proved.

Theorem Let P be a continuous operation defined on a neighborhood Ω of $(x_0, y_0) \in X \times Y$ mapping Ω into Z . Assume:

- (1) $P(x_0, y_0) = 0$
- (2) P'_y exists and is continuous in Ω
- (3)* the operation $P'_y(x_0, y_0) \in [Y \rightarrow Z]$ has a bounded linear inverse

$$\Gamma = [P'_y(x_0, y_0)]^{-1} \in [Z \rightarrow Y]; \|\Gamma\| \leq k_1$$

- (4) there is a function $g_1(u, v)$ non-decreasing in each argument with the other fixed such that

$$(x, y) \in \Omega \Rightarrow \|P'_y(x, y) - P'_y(x_0, y_0)\| \leq g_1(\|x-x_0\|, \|y-y_0\|)$$

- (5) there is a non-decreasing function g_2 such that

$$(x, y) \in \Omega \Rightarrow \|P(x, y_0)\| \leq g_2(\|x-x_0\|)$$

- (6) $S = \{(x, y): \|x-x_0\| < \delta, \|y-y_0\| \leq \varepsilon\} \subset \Omega$

- (7) $\left\{ \begin{array}{l} k_1 g_1(\delta, \varepsilon) \leq \alpha < 1 \\ k_1 g_2(\delta) \leq \varepsilon(1-\alpha) \end{array} \right\}$

* $[Y \rightarrow Z]$ is the space of bounded linear operators mapping Y into Z .

Then an operation F exists defined for $\|x - x_0\| < \delta$ and mapping $\{x \in X: \|x - x_0\| < \delta\}$ into $\{y \in Y: \|y - y_0\| \leq \varepsilon\}$ and possessing the following properties:

$$(a) \quad P(x, F(x)) = 0 \quad (\|x - x_0\| < \delta)$$

$$(b) \quad F(x_0) = y_0$$

$$(c) \quad y = F(x) \text{ depends continuously on } x \text{ for } \|x - x_0\| < \delta$$

Germ of Proof Define $Q^{(x)}(y) = y - \Gamma P(x, y)$. With $(x, y) \in S$ and $\bar{y} \in Y$, $Q^{(x)'}(y)(\bar{y}) = -\Gamma[P'_y(x, y) - P'_y(x_0, y_0)](\bar{y})$. If $\|x - x_0\| < \delta$, $Q^{(x)}(y)$ maps the ball defined by $\|y - y_0\| \leq \varepsilon$ into itself and is a contraction on that ball.

The fixed point of $Q^{(x)}(y)$ is a zero of $P(x, y)$.

Remarks 1) This theorem should be compared with the more standard Banach space implicit function theorem ([1], pp. 686, 687). Beyond any difficulty involved in applying the more standard theorem (finding the partial derivative, inverse, etc.) applying the above theorem is often rather straightforward (bounding functions by non-decreasing functions and satisfying condition (7)). The additional work gives the explicit ε and δ needed to evaluate (IV-1).

2) Further discussion of the implicit function theorem is given in [5].

The next section applies the above theorem to the equation of first variation.

V. EXAMPLE

To fix ideas, we detail here a simple illustrative example, one which is solvable exactly so that an assessment of the method in this case is easily carried out. Consider the scalar differential equation

$$\dot{m}(t) + m(t) + xm^3(t) = 0, \quad m(0) = 1, \quad t \in [0, 1] = T \quad (V-1)$$

where x is a parameter. Let $y(t) = m(t) - 1$ so that $y(0) = 0$ (for a linear space). Then

$$\dot{y}(t) + y(t) + 1 + x(y(t)+1)^3 = 0, \quad y(0) = 0, \quad t \in T \quad (V-2)$$

The solution to (V-2) is

$$y(t) = e^{-t} [1 + x(1 - e^{-2t})]^{-1/2} - 1 \quad t \in T \quad (V-3)$$

If $x_0 = 0$, then

$$y_0(t) = e^{-t} - 1 \quad t \in T \quad (V-4)$$

and the equation of first variation (with $\delta x = x$ since $x_0 = 0$) is

$$\delta \dot{y}(t) = -\delta y(t) - e^{-3t} x, \quad \delta y(0) = 0, \quad t \in T \quad (V-5)$$

the solution to which is

$$\delta y(t) = \frac{x}{2} e^{-t} (e^{-2t} - 1) \quad t \in T \quad (V-6)$$

To use the theorem of Section IV let $z = P(x, y)$ be defined by

$$z(t) = \dot{y}(t) + y(t) + 1 + x(y(t)+1)^3, \quad y(0) = 0, \quad t \in T \quad (V-7)$$

Let the space Y be a space of continuously differentiable functions on T satisfying $y(0) = 0$ and with norm (chosen for convenience)

$$\|y\| = \max_{t \in T} \max \left\{ |e^t y(t)|, \frac{1}{2} |e^t \dot{y}(t)| \right\} \quad (V-8)$$

A bound on $|y(t)|$ can be obtained from the norm of (V-8). First, note that $|y(t)| \leq e^{-t} \|y\|$. Then,

$$|y(t)| \leq \int_0^t |\dot{y}(s)| ds \leq \int_0^t 2\|y\| e^{-s} ds = 2(1 - e^{-t})\|y\| \quad (V-9)$$

$$\max_{t \in T} |y(t)| \leq \max_{t \in T} \min \{e^{-t}, 2(1 - e^{-t})\} \|y\| = \frac{2}{3} \|y\|$$

Let Z be a space of continuous functions with norm

$$\|z\| = \max_{t \in T} |e^t z(t)| \quad (V-10)$$

X is the real line with absolute value for a norm. From (III-3) the operation $z = P'_y(x_0, y_0)y$ is defined by

$$z(t) = \dot{y}(t) + y(t) \quad t \in T \quad (V-11)$$

The inverse operation $y = [P'_y(x_0, y_0)]^{-1} z$ is defined by (see (III-4))

$$y(t) = \int_0^t e^{-(t-s)} z(s) ds \quad t \in T \quad (V-12)$$

To bound $\| [P'_y(x_0, y_0)]^{-1} \|$ first note that from (V-12) we have

$$| e^t y(t) | \leq \int_0^t | e^s z(s) | ds \leq \|z\| \quad t \in T \quad (V-13)$$

From (V-11) and (V-13),

$$| e^t \dot{y}(t) | \leq | e^t z(t) | + | e^t y(t) | \leq 2\|z\| \quad t \in T \quad (V-14)$$

Thus from (V-8) and the definition of a norm of a linear operator,

$$\| [P'_y(x_0, y_0)]^{-1} \| \leq 1 \quad (V-15)$$

where the last norm is the one for $[Z \rightarrow Y]$.

Now the two non-decreasing functions for the theorem must be found. With $\bar{y} \in Y$ and $t \in T$

$$\begin{aligned} e^t [P'_y(x, y) - P'_y(x_0, y_0)] \bar{y}(t) &= e^t 3x[(y(t)+1)]^2 \bar{y}(t) \\ &= 3x[e^t(y(t)+1)]^2 e^{-t} \bar{y}(t) \end{aligned} \quad (V-16)$$

Hence,

$$\begin{aligned} \|P'_y(x, y) - P'_y(x_0, y_0)\| &\leq 3 |x| \max_{t \in T} [e^t(y_0(t)+1) + e^t(y(t)-y_0(t))]^2 \frac{4}{9} \\ &\leq \frac{4}{3} |x-x_0| (1+\|y-y_0\|)^2 \quad (x_0 = 0) \\ &\equiv g_1(|x-x_0|, \|y-y_0\|) \end{aligned} \quad (V-17)^*$$

Also,

$$P(x, y_0)(t) = x(y_0(t)+1)^3 = x e^{-3t} \quad t \in T \quad (V-18)$$

so that

$$\|P(x, y_0)\| = \max_{t \in T} |x e^{-2t}| \leq |x-x_0| \equiv g_2(|x-x_0|) \quad (V-19)$$

* $\max_{t \in T} e^{-t} |\bar{y}(t)| \leq \max_{t \in T} \min \{e^{-2t}, 2e^{-t}(1-e^{-t})\} \|\bar{y}\| = \frac{4}{9} \|\bar{y}\|$

Condition (7) of the theorem becomes

$$\left\{ \begin{array}{l} 4/35(1+\epsilon)^2 \leq \alpha < 1 \\ \delta \leq \epsilon(1-\alpha) \end{array} \right\} \quad (V-20)$$

For $\delta = 0.01$, it is found that (V-20) is satisfied with $\alpha = 0.0136$, $\epsilon = 0.0102$. That is, if $|x| \leq 0.01$, the theorem states that $\|y - y_0\| \leq 0.0102$. From the actual solution, (V-3), one finds that $\|y - y_0\| \leq 0.00432$. Note that the exact information is typically not available.

To use (IV-1), we have

$$e^t \left[\left(P'_x(x, y) - P'_x(x_0, y_0) \right) \bar{x} \right](t) = e^t \left[(y(t)+1)^3 - (y_0(t)+1)^3 \right] \bar{x} \quad (V-21)$$

so that

$$\begin{aligned} & \|P'_x(x, y) - P'_x(x_0, y_0)\| \\ & \leq \max_{t \in T} e^{-t} |y(t) - y_0(t)| \cdot e^{2t} \cdot \left[(y(t)+1)^2 + |y(t)+1| \cdot |y_0(t)+1| + (y_0(t)+1)^2 \right] \\ & \leq \frac{4}{9} \|y - y_0\| \left[(1 + \|y - y_0\|)^2 + (1 + \|y - y_0\|) + 1 \right] \end{aligned} \quad (V-22)$$

Relation (IV-1) becomes

$$\begin{aligned} & \|y - y_0 - F'(x_0)(x - x_0)\| \\ & \leq \sup_{\substack{|x - x_0| < \delta_1 \\ \|y - y_0\| \leq \epsilon_1}} \frac{4}{9} \|y - y_0\| \left[(1 + \|y - y_0\|)^2 + (1 + \|y - y_0\|) + 1 \right] |x - x_0| \\ & \quad + \sup_{\substack{|x - x_0| < \delta_2 \\ \|y - y_0\| \leq \epsilon_2}} \frac{4}{3} |x - x_0| (1 + \|y - y_0\|)^2 \|y - y_0\| \\ & = 1.35(10)^{-4} \end{aligned} \quad (V-23)$$

The actual bound on $\|y - y_0 - F'(x_0)(x - x_0)\|$ (from (V-2) through (V-6) and using the norm of (V-8)) is $0.281(10)^{-4}$. We could improve (V-23) if we used the actual $\epsilon (= 0.00432)$ but this information is usually not available (which is, of course, the motivation of this paper). With the actual bounds unavailable (as is usually the case) the analysis might be interpreted to show the linearization to be accurate in the sense that the error in linearization

$(\|y - y_0 - F'(x_0)(x - x_0)\|)$ is substantially smaller than the variations in the solution (measured by ε).

REFERENCES

1. L. V. Kantorovich and G. P. Akilov, "Functional Analysis and Normed Linear Spaces", (translation from Russian), The Macmillan Company, New York (1964).
2. H. S. Witsenhausen, "On the Sensitivity of Optimal Control Systems", I.E.E.E. Transactions on Automatic Control, Vol. AC-10, No. 10, pp. 495-496 (October 1965).
3. C. A. Desoer and K. K. Wong, "Small Signal Behavior of Nonlinear Lumped Networks", Proc. I.E.E.E., Vol. 56, No. 1, pp. 14-22 (January 1968).
4. R. S. Rutman and P. V. Kokotović, "On the Generalization of Sensitivity Equations", Proc. 3rd Annual Allerton Conf. on Circuit and System Theory, pp. 27-34 (1965).
5. J. M. Holtzman, "Explicit ε and δ for the Implicit Function Theorem", to appear.

GENERATION OF SENSITIVITY FUNCTIONS FOR LINEAR SYSTEMS USING LOW ORDER MODELS

DENNIS F. WILKIE
Transportation Research and
Planning Division
Ford Motor Company
Dearborn, Michigan

and

WILLIAM R. PERKINS
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois

ABSTRACT

New proofs are given for the recently demonstrated Total Symmetry and Complete Simultaneity Properties for the Companion canonic form for single-input, linear, time invariant, controllable systems. These proofs result in a convenient closed form expression for the Complete Simultaneity Property. The use of these properties to generate by one nth order sen-

sitivity model all the sensitivity functions $\frac{\partial x_i}{\partial v_j} \bigg|_{y_0}$, $i=1, \dots, n$, $j=1, \dots, r$,

for a single-input, linear, time invariant, controllable nth order system which depends on r different parameters is reviewed. This method represents an improvement over known methods for generating the sensitivity functions, which generally require a composite dynamic system of order $n(r+1)$. This result is then extended to the case of multi-input, normal linear systems, where at most $2m-1$ dynamic nth order systems are needed in addition to the system to generate all the sensitivity functions of the system state with respect to any number of parameters (m is the dimension of $\underline{u}$). It is shown that the algebraic calculations that must be made in the m input case are much less than m times the calculations needed for the single-input case. The implications of these results for the computer aided sensitivity analysis of systems are discussed.

INTRODUCTION

It has been demonstrated recently that for a single-input, linear, time invariant controllable nth order system the sensitivity functions (or parameter influence coefficients) of the state with respect to any number of system parameters can be generated by one nth order sensitivity model, in addition to a system model [1,2]. The use of previously available techniques [3-5] to generate the sensitivity functions of the state of a system described by the state equations

$$\dot{\underline{x}}(t, \underline{v}) = A(\underline{v}) \underline{x}(t, \underline{v}) + \underline{b}(\underline{v}) u(t) \quad (1)$$

where

$\underline{x}$ = n dimensional state vector

$\underline{v}$ = r dimensional parameter vector

u = scalar input

$A(\underline{v})$ = $n \times n$ constant matrix

$\underline{b}(\underline{v})$ = $n \times 1$ vector

requires in general the use of r sensitivity models of order n in addition to the system itself. Thus, the order of the system together with the sensitivity models is $n(r+1)$, and this order becomes very large for high order systems containing many variable parameters. The use of unnecessarily high order dynamic systems to generate sensitivity functions [4] is undesirable for several reasons. For example, when sensitivity functions

are generated for use in self-optimizing and adaptive control systems [5-7] using a digital computer for system simulation, the numerical integration of the system equations is usually the most time consuming aspect of the analysis. Furthermore, when using analog simulation, the amount of analog equipment needed for simulating a large system with its sensitivity models may be prohibitive. The "sensitivity points" method [5] does not generate such high order models, but its application is limited to systems which can be represented in a particular form and in which the sensitivity of a scalar output is desired.

The result of [1,2] was based on the development of two previously unnoticed properties of the sensitivity functions of the companion canonic form of single-input, linear, time invariant controllable systems. These are the Total Symmetry and Complete Simultaneity Properties. The discussion of these properties in [1,2] depended on systematic inspection of block diagrams of the system and sensitivity models and a particular case of the sensitivity points method [5]. In this paper, new proofs of these properties are given which do not require such observations from block diagrams and do not utilize the sensitivity points method. These proofs yield a closed form statement of the Complete Simultaneity Property which was not given in [1,2]. In addition the result of [1] is extended to the case of multi-input normal systems. It is shown that the sensitivity functions of the state of normal systems described by

$$\dot{\underline{z}}(t, \underline{v}) = A(\underline{v}) \underline{z}(t, \underline{v}) + B(\underline{v}) \underline{u}(t), \quad (2)$$

where $\underline{u}(t)$ is an m vector and all other quantities are as in (1), can be generated using at most $2m-1$ dynamic n th order models in addition to the system itself. As for single-input systems, the number of sensitivity models for this class of multi-input systems is independent of the number of system parameters. Further, all $2m-1$ models are seen to be identical in form, and it is shown that the algebraic calculations needed to apply these results to m input systems are much less than m times the number of calculations needed for the single input case.

THE TOTAL SYMMETRY AND COMPLETE SIMULTANEITY PROPERTIES

Consider the single input, linear, time invariant system described by

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_{n-1} &= z_n \\ \dot{z}_n &= - \sum_{i=1}^n \alpha_i z_i + u, \quad z_i(0) = 0. \end{aligned} \quad (3)$$

This will be called the companion form system. The parameters α_i , $i=1, \dots, n$ are the coefficients of the characteristic equation of the system. Some very interesting properties of the partial derivatives of the states of this system with respect to the alpha parameters $\left. \frac{\partial z_i}{\partial \alpha_j} \right|_{\underline{z}_0}^*$, $i, j=1, \dots, n$, can

* Since the sensitivity functions are by definition evaluated at some nominal parameter value $\underline{z}^0$, the subscript $\underline{z}^0$ on $\left. \frac{\partial z}{\partial \alpha_i} \right|_{\underline{z}^0}$ and similar subscripts on other derivatives are omitted in the remainder of the paper.

be demonstrated. These are the following:

Property I: The Total Symmetry Property

Define a sensitivity vector as

$$\xi_i \triangleq \frac{\partial z}{\partial \alpha_i}$$

and a sensitivity matrix $[\xi_{ij}]$ as

$$[\xi_{ij}] \triangleq [\xi_1 \xi_2 \dots \xi_n]. \quad (4)$$

Then the sensitivity matrix $[\xi_{ij}]$ has the following Total Symmetry Property,

$$\xi_{i,j}(t) = \xi_{i-1,j+1}(t), \quad i=2, \dots, n, \quad j=1, \dots, n-1, \quad (5)$$

so that all elements along the anti-diagonals of the $[\xi_{ij}]$ matrix are equal as shown in (6),

$$[\xi_{i,j}] = \begin{bmatrix} \xi_{11} & \xi_{12} & \xi_{13} & \dots & \xi_{1n} \\ \xi_{12} & \xi_{13} & & & \xi_{2n} \\ \xi_{13} & & & & \vdots \\ \vdots & & & & \xi_{n-1,n} \\ \xi_{1n} & \xi_{2n} & & \xi_{n-1,n} & \xi_{n,n} \end{bmatrix}. \quad (6)$$

Proof:

Differentiating (3) with respect to any α_j , it follows that

$$\begin{aligned} \left(\frac{\partial \dot{z}_1}{\partial \alpha_j} \right) &= \frac{\partial z_2}{\partial \alpha_j} \\ \left(\frac{\partial \dot{z}_2}{\partial \alpha_j} \right) &= \frac{\partial z_3}{\partial \alpha_j} \\ &\vdots \\ \left(\frac{\partial \dot{z}_{n-1}}{\partial \alpha_j} \right) &= \frac{\partial z_n}{\partial \alpha_j} \\ \left(\frac{\partial \dot{z}_n}{\partial \alpha_j} \right) &= - \sum_{i=1}^n \alpha_i \frac{\partial z_i}{\partial \alpha_j} - z_j, \quad \frac{\partial z(o)}{\partial \alpha_j} = 0, \end{aligned} \quad (7)$$

since the conditions for interchanging the order of differentiation with respect to time and parameters are met. Similarly, differentiating (3) with respect to α_{j+1} , it follows that

$$\begin{aligned} \left(\frac{\partial \dot{z}_1}{\partial \alpha_{j+1}} \right) &= \frac{\partial z_2}{\partial \alpha_{j+1}} \\ \left(\frac{\partial \dot{z}_2}{\partial \alpha_{j+1}} \right) &= \frac{\partial z_3}{\partial \alpha_{j+1}} \\ &\vdots \end{aligned}$$

$$\begin{aligned} \left(\frac{\partial \dot{z}_{n-1}}{\partial \alpha_{j+1}} \right) &= \frac{\partial z_n}{\partial \alpha_{j+1}} \\ \left(\frac{\partial \dot{z}_n}{\partial \alpha_{j+1}} \right) &= - \sum_{i=1}^n \alpha_i \frac{\partial z_i}{\partial \alpha_{j+1}} - z_{j+1}, \quad \frac{\partial z(0)}{\partial \alpha_{j+1}} = 0. \end{aligned} \quad (8)$$

Equations (7) and (8) are the sensitivity equations for the system (3) with respect to α_j and α_{j+1} respectively. The initial conditions for (7) and (8) are zero since $\dot{z}(0) = 0$, independent of the parameters α_j . Integration of (8) with respect to time yields

$$\int_0^t \left(\frac{\partial \dot{z}_n}{\partial \alpha_{j+1}} \right) d\tau = \frac{\partial z_n}{\partial \alpha_{j+1}} = - \sum_{i=1}^n \alpha_i \int_0^t \frac{\partial z_i}{\partial \alpha_{j+1}} d\tau - \int_0^t z_{j+1} d\tau. \quad (9)$$

Substituting the first $n-1$ equations of (8), (9) yields

$$\left(\frac{\partial \dot{z}_{n-1}}{\partial \alpha_{j+1}} \right) = -\alpha_1 \int_0^t \frac{\partial z_1}{\partial \alpha_{j+1}} d\tau - \sum_{i=2}^n \alpha_i \frac{\partial z_{i-1}}{\partial \alpha_{j+1}} - z_j, \quad j=1, \dots, n-1. \quad (10)$$

Now define

$$\begin{aligned} w_1(t) &= \int_0^t \frac{\partial z_1}{\partial \alpha_{j+1}} d\tau, \\ w_2(t) &= \dot{w}_1(t) = \frac{\partial z_1}{\partial \alpha_{j+1}} \\ w_3(t) &= \frac{\partial z_2}{\partial \alpha_{j+1}} \\ &\vdots \\ w_n(t) &= \frac{\partial z_{n-1}}{\partial \alpha_{j+1}}. \end{aligned} \quad (11)$$

Then, substituting (11) in (8) and replacing the last equation of (8) with (10),

$$\begin{aligned} \dot{w}_1(t) &= w_2(t) \\ \dot{w}_2(t) &= w_3(t) \\ &\vdots \\ \dot{w}_{n-1}(t) &= w_n(t) \\ \dot{w}_n(t) &= - \sum_{i=1}^n \alpha_i w_i(t) - z_j(t), \quad w(0) = 0. \end{aligned} \quad (12)$$

Comparing (12) and (7) and invoking the uniqueness theorem for the solutions of differential equations, it is evident that

$$w(t) = \frac{\partial \tilde{z}(t)}{\partial \alpha_j}. \quad (13)$$

Thus,

$$w_1(t) = \int_0^t \frac{\partial z_1}{\partial \alpha_{j+1}} d\tau = \frac{\partial z_1}{\partial \alpha_j} \quad (14a)$$

and

$$w_k(t) = \frac{\partial z_{k-1}}{\partial \alpha_{j+1}} = \frac{\partial z_k}{\partial \alpha_j}, \quad k=2, \dots, n, \quad j=1, \dots, n-1. \quad (14b)$$

The result of (14b) proves Property I.

The result of (14a) stated differently is

$$\frac{\partial z_1}{\partial \alpha_{j+1}} = \left(\frac{\partial \dot{z}_1}{\partial \alpha_j} \right), \quad j=1, \dots, n-1. \quad (15)$$

This, along with (7) for $j=1$ yields a special case of the previously known sensitivity points method [4] for generating the sensitivity functions

$\frac{\partial z_1}{\partial \alpha_j}$, $j=1, \dots, n$, which is shown in Figure 1.

The second result regarding the generation of the sensitivity functions for a companion form system follows from Property I.

Property II: The Complete Simultaneity Property

All the sensitivity functions $\frac{\partial z_1}{\partial \alpha_j}$, $i, j=1, \dots, n$ can be obtained as linear combinations of the signals appearing on the one sensitivity model of the system shown in Figure 1, together with the system states.

Proof:

As proved in Property I and shown in Figure 1, $\frac{\partial z_1}{\partial \alpha_1}, \dots, \frac{\partial z_1}{\partial \alpha_n}$ can be obtained from one sensitivity model of the system. Now, from (3),

$$\left(\frac{\partial \dot{z}_1}{\partial \alpha_n} \right) = \frac{\partial z_2}{\partial \alpha_n}. \quad (16)$$

But from Property I, $\frac{\partial z_1}{\partial \alpha_n} = \frac{\partial z_n}{\partial \alpha_1}$. Therefore

$$\frac{\partial z_2}{\partial \alpha_n} = \left(\frac{\partial \dot{z}_n}{\partial \alpha_1} \right) = -z_1 - \sum_{i=1}^n \alpha_i \frac{\partial z_i}{\partial \alpha_1} = -z_1 - \sum_{i=1}^n \alpha_i \frac{\partial z_1}{\partial \alpha_i}. \quad (17)$$

Then

$$\frac{\partial z_3}{\partial \alpha_n} = \left(\frac{\partial \dot{z}_2}{\partial \alpha_n} \right) = -\dot{z}_1 - \sum_{i=1}^n \alpha_i \left(\frac{\partial \dot{z}_1}{\partial \alpha_1} \right) = -z_2 - \sum_{i=1}^{n-1} \alpha_i \frac{\partial z_{i+1}}{\partial \alpha_1} - \alpha_n \frac{\partial z_2}{\partial \alpha_n}, \quad (18)$$

and further

$$\frac{\partial z_4}{\partial \alpha_n} = \left(\frac{\partial \dot{z}_3}{\partial \alpha_n} \right) = -\dot{z}_2 - \sum_{i=1}^{n-1} \alpha_i \left(\frac{\partial \dot{z}_{i+1}}{\partial \alpha_1} \right) - \alpha_n \left(\frac{\partial \dot{z}_2}{\partial \alpha_n} \right) \quad (19)$$

which is equivalent to

$$\frac{\partial z_4}{\partial \alpha_n} = -z_3 - \sum_{i=1}^{n-2} \alpha_i \frac{\partial z_{i+2}}{\partial \alpha_1} - \alpha_{n-1} \frac{\partial z_2}{\partial \alpha_n} - \alpha_n \frac{\partial z_3}{\partial \alpha_n}. \quad (20)$$

Continuation of this procedure yields

$$\frac{\partial z_4}{\partial \alpha_n} = -z_{n-1} - \sum_{i=1}^2 \alpha_i \frac{\partial z_{i+n-2}}{\partial \alpha_1} - \alpha_3 \frac{\partial z_2}{\partial \alpha_n} - \alpha_4 \frac{\partial z_3}{\partial \alpha_n} \dots - \alpha_{n-1} \frac{\partial z_{n-2}}{\partial \alpha_n} - \alpha_n \frac{\partial z_{n-1}}{\partial \alpha_n}, \quad (21)$$

and in general

$$\frac{\partial z_k}{\partial \alpha_n} = -z_{k-1} - \sum_{i=1}^{n+2-k} \alpha_i \frac{\partial z_{i+k-2}}{\partial \alpha_1} - \sum_{i=n+3-k}^n \alpha_i \frac{\partial z_{i-1}}{\partial \alpha_n}, \quad k \geq 3. \quad (22)$$

Thus, $\frac{\partial z_2}{\partial \alpha_n}, \dots, \frac{\partial z_n}{\partial \alpha_n}$ can be generated as linear combinations of

$\frac{\partial z_1}{\partial \alpha_1}, \dots, \frac{\partial z_n}{\partial \alpha_1}$ with $z_1, \dots, z_{n-1}$.

Properties I and II demonstrate that at most $2n-1$ signals are needed to obtain all the elements of the matrix $[\xi_{ij}]$. Of these signals, $n+1$ are obtained directly as the signals at nodes 1 through $n+1$ of the sensitivity model as shown in Figure 1. The remaining $n-2$ functions are obtained as linear combinations of the signals at nodes 1 through $n+1$ of the sensitivity model with the states $z_1, \dots, z_{n-1}$ as given by (22). The generation of the sensitivity functions using these results for a third order system are shown in Figure 2. In the next section, the application of these results to systems which are not in companion form is considered.

APPLICATION TO A GENERAL SYSTEM

The results given in this section were previously presented in [1,2]. However, they are reproduced here for completeness.

In this section, a method of utilizing Properties I and II for systems in companion form in the sensitivity analysis of general single-input, linear, time-invariant controllable systems is developed. It has been shown [2-13] that for a controllable system described by the state equations

$$\begin{aligned}\dot{\underline{x}} &= A\underline{x} + \underline{b}u \\ \underline{y} &= C\underline{x}\end{aligned}\quad (23)$$

where $\underline{x}$, A , $\underline{b}$ and u are as defined in (1), $\underline{y}$ is a p dimensional output vector, and C a $p \times n$ matrix, that a non-singular transformation

$$\underline{x} = T\underline{z} \quad (24)$$

exists such that

$$\dot{\underline{z}} = \hat{A}\underline{z} + \hat{\underline{b}}u \quad (25)$$

where

$$\hat{A} = T^{-1}AT = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ -\alpha_1 & -\alpha_2 & & -\alpha_{n-1} & -\alpha_n \end{bmatrix}$$

and

$$\hat{\underline{b}} = T^{-1}\underline{b} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

The elements $\alpha_1, \alpha_2, \dots, \alpha_n$ in the $\hat{A}$ matrix are the coefficients in the characteristic equation

$$\lambda^n + \alpha_n \lambda^{n-1} + \dots + \alpha_2 \lambda + \alpha_1 = 0.$$

The form of equations (25) is exactly the same as (3). If the A and $\underline{b}$ matrices of (23) depend on a parameter vector $\underline{v}$ as in (1), then it is evident that the transformation T in (24) will depend on $\underline{v}$ in general, i.e.,

$$T = T(\underline{v}). \quad (26)$$

Further, the coefficients of the characteristic equations $\alpha_1, \dots, \alpha_n$ will also depend on $\underline{v}$. For convenience, define the vector $\underline{\alpha}$ as

$$\underline{\alpha}(\underline{v}) \triangleq \begin{bmatrix} \alpha_1(\underline{v}) \\ \alpha_2(\underline{v}) \\ \vdots \\ \alpha_n(\underline{v}) \end{bmatrix}. \quad (27)$$

Now with the system (1) in the canonical form (25), it is evident that $\underline{z}$ represents a new parameter vector (n dimensional in the canonic system, and that no other parameters exist in this system to affect the state $\underline{z}$. The parameters $\alpha_1, \dots, \alpha_n$ will be referred to as essential parameters.

An important result which follows from this discussion and Properties I and II of the previous section is:

Assertion 1:

The sensitivity functions of all states x_i in the system (1) with respect to all r parameters ($r \leq n$) can be obtained as linear combinations of the sensitivity functions $\frac{\partial z_i}{\partial \alpha_j}$, $i, j=1, \dots, n$, of the canonic system and the states $\underline{z}$.

Proof:

This follows directly from (3) and (16). For any parameter v_i ,

$$\frac{\partial \underline{x}}{\partial v_i} = \frac{\partial T}{\partial v_i} \underline{z} + T(\underline{v}) \frac{\partial \underline{z}}{\partial \alpha_j} \frac{\partial \alpha_j}{\partial v_i} = \frac{\partial T}{\partial v_i} T^{-1} \underline{x} + T(\underline{v}) \frac{\partial \underline{z}}{\partial \alpha_j} \frac{\partial \alpha_j}{\partial v_i}. \quad (28)$$

At this point, by combining Properties I and II with Assertion 1, the following important result is obtained:

Assertion 2:

Consider a linear, time-invariant, single input controllable system described by the equations

$$\begin{aligned} \dot{\underline{x}} &= A(\underline{v})\underline{x} + \underline{b}(\underline{v})u \\ \underline{y} &= C(\underline{v})\underline{x}. \end{aligned}$$

The sensitivity functions of all states (or outputs) with respect to all r parameters can be generated by algebraically combining the signals on a single n th order canonical sensitivity model in addition to the considered system. This is accomplished by combining the results of Properties I and II and Assertion 1 as shown in Figure 3.

In order to realize the utility of this result, suppose the sensitivity functions of the states of a linear, single input, time-invariant, controllable twentieth order system with respect to six variable parameters are desired. By straightforward use of currently-available methods, this would require 6 twentieth order sensitivity models in addition to the system itself, i.e., a 140th order dynamic system. However, using the techniques of this paper, only one twentieth order system would be needed in addition to the given system, i.e., an 40th order system.

Indeed this represents a considerable savings.

To implement the results of this section in a computer simulation, the transformation T of (26) as well as $\frac{\partial T}{\partial v_i}$ and $\frac{\partial \alpha_k}{\partial v_i}$, $k=1, \dots, n$, $i=1, \dots, r$ must be obtained. If it were necessary to find T and the α_k 's as general functions of $\underline{v}$ to compute these necessary derivatives, the utility of Assertion 2 would be questionable. However, it has been shown [1,2] that $\frac{\partial T}{\partial v_i}$ and $\frac{\partial \alpha_k}{\partial v_i}$ at any $\underline{q} \in Q$ can be obtained recursively by an extension of the Leverrier algorithm. These results are reviewed in the Appendix.

EXTENSION TO MULTI-INPUT SYSTEMS

The results of the previous sections can be extended to a class of linear, time-invariant, multi-input controllable systems [2]. In this case, the system is described by the state equations

$$\begin{aligned}\dot{\underline{x}} &= A(\underline{v})\underline{x} + B(\underline{v})\underline{u}, \\ \underline{y} &= C(\underline{v})\underline{x},\end{aligned}\tag{29}$$

where all quantities are as in (1) and (23) except $\underline{u}$ which is an m dimensional input vector. Let the following assumptions be made:

- (i) Initial conditions in the system are zero.
- (ii) The system is normal, i.e. completely controllable from any one input acting along. This means the pairs (A, b_i) , $i=1, \dots, m$ are all controllable.

The first of these assumptions must also be made in the previous results for single input systems. However, in the multi-input case, this restriction is less severe since arbitrary initial conditions can be achieved with an additional input to the system, whose corresponding b vector will have components depending on initial conditions. In addition, this means that other system parameters may depend on initial conditions, which are considered as n more parameters in this case.

For a linear system with zero initial conditions, it follows from the principle of superposition that

$$\underline{x} = \underline{x}^1 + \underline{x}^2 + \dots + \underline{x}^m\tag{30}$$

where $\underline{x}^1$ is the component of the response due to u_1 acting alone, $\underline{x}^2$ is that due to u_2 , etc. By assumption (ii), transformations $T_1, \dots, T_m$ exist which would transform the corresponding single input systems (with inputs $u_1, \dots, u_m$) into companion form. Now since

$$\frac{\partial \underline{x}}{\partial v_i} = \frac{\partial \underline{x}^1}{\partial v_i} + \frac{\partial \underline{x}^2}{\partial v_i} + \dots + \frac{\partial \underline{x}^m}{\partial v_i}, \quad i=1, \dots, r\tag{31}$$

it is evident that the desired sensitivity functions are simply obtained if $\frac{\partial \underline{x}^1}{\partial v_i}, \dots, \frac{\partial \underline{x}^m}{\partial v_i}$ are known. But from the results of the previous sections it follows that only one canonic sensitivity model is required for each $\underline{x}^k$

to generate $\frac{\partial \tilde{x}^k}{\partial v_i}$. Thus, m companion sensitivity models are needed to generate the sensitivity functions $\frac{\partial \tilde{x}}{\partial v_i}$, assuming that $\tilde{z}^1, \dots, \tilde{z}^m$ can be calculated. This is equivalent to determining $\tilde{x}^1, \dots, \tilde{x}^m$ since

$$\tilde{z}^1 = (T_1)^{-1} \tilde{x}^1, \dots, \tilde{z}^m = (T_m)^{-1} \tilde{x}^m. \quad (32)$$

The only way of obtaining $\tilde{x}^1, \dots, \tilde{x}^m$ simultaneously is by using $m-1$ system models with scalar inputs $u_1, \dots, u_m$ in addition to the system itself. Since $m-1$ system models must be used, and only $\tilde{z}^1, \dots, \tilde{z}^m$ are required (as opposed to $\tilde{x}^1, \dots, \tilde{x}^m$), it is best to use $m-1$ companion system models. Further, since the only parameters in the companion models are $\alpha_1, \dots, \alpha_n$ (the coefficients of the characteristic equation), it is evident that all the companion system models are identical. Thus, for a normal system of order n , all the sensitivity functions $\frac{\partial x_i}{\partial v_j}$, $i=1, \dots, n, j=1, \dots, r$, can be generated using at most $2m-1$ companion models of order n in addition to the system itself, as shown in Figure 4.

Important points to note in comparing this result with the single input results obtained previously are the following:

- (1) All of the companion system and sensitivity models are decoupled, have a simple structure, and are identical. This facilitates the analog or digital simulation of the composite system.
- (2) Since all of the $\alpha_1, \dots, \alpha_n$ are the same for every model, they need be calculated only once as in the single input case. This also is true for $\frac{\partial \alpha}{\partial v_i}$, $i=1, \dots, r$.
- (3) By using companion models of the system, $\tilde{z}^2, \dots, \tilde{z}^m$ are generated directly and hence $T_2, \dots, T_m$ need not be inverted. Thus the number of matrix inversions is the same as in the single input case.
- (4) The only additional algebraic calculations for the multi-input case are those of calculating $T_2, \dots, T_m$ and $\frac{\partial T_2}{\partial v_1}, \dots, \frac{\partial T_m}{\partial v_1}$, $i=1, \dots, r$.
- (5) The number of required models ($2m-1$) is independent of the number of parameters, as is the structure of the models.

Now assuming that the number of parameters to be adjusted in an n th order system would be at least n , and that generally $2m-1 < n$, the approach of this section could result in a considerable savings of simulation time (or in the analog case an equipment savings). This would be especially true for high order systems with only a few inputs. These results could be further generalized to non-normal systems which have real eigenvalues and are diagonalizable, but the procedure becomes quite complicated in that case and would probably not result in substantial savings. The techniques of this section are applied to the multi-input system considered in the following example.

Example

The equation of the longitudinal motion of a jet transport can be expressed in the following normal form [18]

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} v_{10} + v_{12} & \frac{v_{10}}{V_T}(g v_1 + V_T v_6) & \frac{v_{10}}{V_T}(v_1 v_4 + v_5) + v_n & v_9 + v_1 v_{10} v_2 + v_{10} v_7 \\ 1 & 0 & 0 & 0 \\ 0 & -g & -V_4 & -V_T v_2 \\ 1 & (g v_1 + v_6 V_T)/V_T & (v_1 v_4 + v_5)/V_T & v_1 v_2 + v_7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} v_1 v_3 v_{10} & v_8 v_{10} + v_{13} \\ 0 & 0 \\ -v_3 V_T & 0 \\ v_1 v_3 & v_8 \end{bmatrix} \begin{bmatrix} \Delta \delta_T \\ \Delta \delta_e \end{bmatrix}, \quad (33)$$

where

- x_1 = incremental pitching velocity
- x_2 = incremental pitch angle
- x_3 = incremental angle of attack
- x_4 = incremental change in air speed
- $\Delta \delta_T$ = incremental change in thrust
- $\Delta \delta_e$ = incremental elevator deflection

and the parameters $v_1, \dots, v_{13}$ are various aerodynamic derivatives. Suppose it is desired to study the sensitivity of the states of the system to changes in these parameters by generating the sensitivity functions $\frac{\partial x_i}{\partial v_j}$, $i=1, \dots, 4$, $j=1, \dots, 13$. These sensitivity functions could be used in an automatic parameter optimization for example. By previous methods, the simultaneous generation of these sensitivity functions would require 13 fourth order sensitivity models in addition to the system itself, that is, a 56th order dynamic system. Using the methods of this paper, only 3 fourth order models in addition to the system, i.e. a 16th order dynamic system, is required. The 56th and 16th order systems were simulated on a digital computer, and the time required to generate the sensitivity functions using the 56th order system was $2\frac{1}{2}$ times that required using the 16th order system. In the case of analog simulation of the sensitivity functions, the equipment savings would be considerable. This time or equipment savings demonstrates the utility of the results of the paper.

CONCLUSIONS

New proofs of the Total Symmetry and Complete Simultaneity Properties of the sensitivity matrix of the companion canonic form for single-input, linear, time invariant systems are given. These proofs do not depend on observations of block diagrams and use of the sensitivity points method as did those in [1,2]. Furthermore, these proofs yield a closed form statement of the Complete Simultaneity property which is more convenient than the statement given in [1,2]. The use of these results in generating with only one sensitivity model all the sensitivity functions $\frac{\partial x_i}{\partial v_j}$, $i=1, \dots, n$, $j=1, \dots, r$, for any single-input, linear, time invariant, controllable nth order system which depends on r parameters is reviewed.

This result is then extended to the case of multi-input, normal linear systems, where at most $2m-1$ dynamic n th order systems are needed in addition to the system to generate all the sensitivity functions of the system state with respect to any number of parameters (m is the dimension of y). It is shown that the algebraic calculations that must be made in the m input case are much less than m times the calculations needed for the single-input case. The implications of these results for the computer aided sensitivity analysis of systems are discussed.

ACKNOWLEDGMENT

The authors acknowledge helpful discussions with Professors J. B. Cruz, Jr. and P. Kokotović.

This work was supported by the Joint Services Electronics Program (U. S. Army, U. S. Air Force, and U. S. Navy) under Contract No. DAAB-07-67-C-0199, and in part by the Air Force under Grant No. 68-1579.

REFERENCES

1. D. F. Wilkie and W. R. Perkins, "Essential Parameters in Sensitivity Analysis," Proc. of the Second IFAC Symposium on System Sensitivity and Adaptivity, Dubrovnik, Yugoslavia, August, 1968.
2. D. F. Wilkie, "Use of the Companion Transformation in Parameter Optimization and Adaptive Control," Report R-386, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois, July, 1968.
3. P. V. Kokotović and R. S. Rutman, "Sensitivity of Automatic Control Systems (Survey)," Automation and Remote Control, No. 4, April, 1965, Moscow, pp. 727-749.
4. R. Tomović, Sensitivity Analysis of Dynamic Systems, McGraw-Hill, 1964, New York.
5. P. V. Kokotović, Method of Sensitivity Points in the Investigation and Optimization of Linear Control Systems," Automation and Remote Control, No. 12, December, 1964, Moscow, pp. 1512-1518.
6. P. V. Kokotović, et al., "Sensitivity Method in the Experimental Design of Adaptive Control Systems," Third IFAC Congress, London, June, 1966, pp. 45B.1-45B.12.
7. S. P. Bingulac, "On the Compatibility of Adaptive Controllers," Fourth Annual Allerton Conference on Circuit and System Theory, October, 1966, University of Illinois, Urbana, Illinois, pp. 8-16.
8. S. P. Bingulac, "A Transformation Improving Stability of Adaptive Systems," Fourth Annual Allerton Conference on Circuit and System Theory, October, 1966, University of Illinois, Urbana, Illinois, pp. 28-35.
9. W. M. Wonham and C. D. Johnson, "Optimal Bang-Bang Control with Quadratic Performance Index," Proc. 1963 4th JACC, pp. 101-112.

10. C. D. Johnson and W. M. Wonham, "A Note on the Transformation to Canonical (Phase-Variable) Form," IEEE Trans. on Automatic Control, Vol. AC-9, pp. 312-313, July, 1964.
11. W. G. Tuel, Jr., "On the Transformation to (Phase-Variable) Canonical Form," IEEE Trans. on Automatic Control, Vol. AC-11, No. 3, July, 1966, p. 607.
12. D. S. Rane, "A Simplified Transformation to (Phase-Variable) Canonical Form," IEEE Trans. on Automatic Control, Vol. AC-11, No. 3, July, 1966, p. 608.
13. C. D. Johnson and W. M. Wonham, "Another Note on the Transformation to Canonical (Phase-Variable) Form," IEEE Trans. on Automatic Control, Vol. AC-11, No. 3, July, 1966, p. 609.
14. R. W. Bass and I. Gura, "High Order System Design Via State Space Considerations," Proc. 1965 JACC, New York, pp. 311-317.
15. L. A. Zadeh and C. Desoer, Linear System Theory, McGraw-Hill, 1963, pp. 300-306.
16. V. N. Fadeeva, Computational Methods of Linear Algebra, Dover, New York, 1969, pp. 177-178.
17. B. S. Morgan, Jr., "Sensitivity Analysis and Synthesis of Multi-Variable Systems," IEEE Trans. on Automatic Control, Vol. AC-11, No. 3, July, 1966, pp. 506-512.
18. E. G. Rynaski and R. F. Whitbeck, "The Theory and Application of Linear Optimal Control," Tech. Rep. AFFOL-TR-65-28, Air Force Flight Dynamics Lab., Research and Technology Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, January, 1966, pp. 145-146.

APPENDIX

The algorithm for computing the transformation to companion form for a given system is as follows [9-13]:

$$T = [\tilde{t}_1 \dots \tilde{t}_n]$$

where

$$\begin{aligned} \tilde{t}_n &= b \\ \tilde{t}_{n-1} &= A\tilde{t}_n + \alpha_n \tilde{t}_n \\ \tilde{t}_{n-2} &= A\tilde{t}_{n-1} + \alpha_{n-1} \tilde{t}_n \\ &\vdots \\ \tilde{t}_1 &= A\tilde{t}_2 + \alpha_2 \tilde{t}_n. \end{aligned} \tag{A.1}$$

The coefficients of the characteristic equation which are required to find the transformation can be calculated by using Leverrier's algorithm [14-17]:

$$\begin{aligned}\alpha_{n+1} &= 1 & S_{n+1} &= I & (\text{nxn identity matrix}) \\ \alpha_{n-j+1} &= -(1/j) \operatorname{tr}(AS_{n-j+2}), & S_{n-j+1} &= \alpha_{(n-j+1)} I + AS_{n-j+2}.\end{aligned}\quad (\text{A.2})$$

A check on the numerical calculations is that $S_1 = 0$ should be obtained.

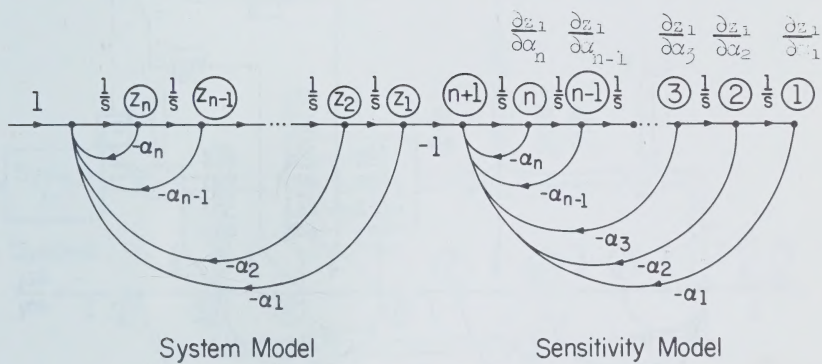
Using these results, one can obtain the algorithm for computing $\frac{\partial T(\underline{y})}{\partial v_i}$, $i=1, \dots, r$, which is as follows [1,2]:

$$\begin{aligned}\frac{\partial T}{\partial v_i} &= \begin{bmatrix} \frac{\partial \tilde{t}_1}{\partial v_i} & \dots & \frac{\partial \tilde{t}_n}{\partial v_i} \end{bmatrix} \\ \frac{\partial \tilde{t}_n}{\partial v_i} &= \frac{\partial \tilde{b}}{\partial v_i} \\ \frac{\partial \tilde{t}_{n-1}}{\partial v_i} &= \frac{\partial A}{\partial v_i} \tilde{t}_n + A \frac{\partial \tilde{t}_n}{\partial v_i} + \frac{\partial \alpha_n}{\partial v_i} \tilde{b} + \alpha_n \frac{\partial \tilde{b}}{\partial v_i} \\ \frac{\partial \tilde{t}_{n-2}}{\partial v_i} &= \frac{\partial A}{\partial v_i} \tilde{t}_{n-1} + A \frac{\partial \tilde{t}_{n-1}}{\partial v_i} + \frac{\partial \alpha_{n-1}}{\partial v_i} \tilde{b} + \alpha_{n-1} \frac{\partial \tilde{b}}{\partial v_i} \\ &\vdots \\ \frac{\partial \tilde{t}_1}{\partial v_i} &= \frac{\partial A}{\partial v_i} \tilde{t}_2 + A \frac{\partial \tilde{t}_2}{\partial v_i} + \frac{\partial \alpha_2}{\partial v_i} \tilde{b} + \alpha_2 \frac{\partial \tilde{b}}{\partial v_i},\end{aligned}\quad (\text{A.3})$$

and the derivatives $\frac{\partial \alpha_k}{\partial v_i}$, $k=1, \dots, n$, $i=1, \dots, r$, can be pointwise calculated by the derivative of the Leverrier algorithm [1,2]:

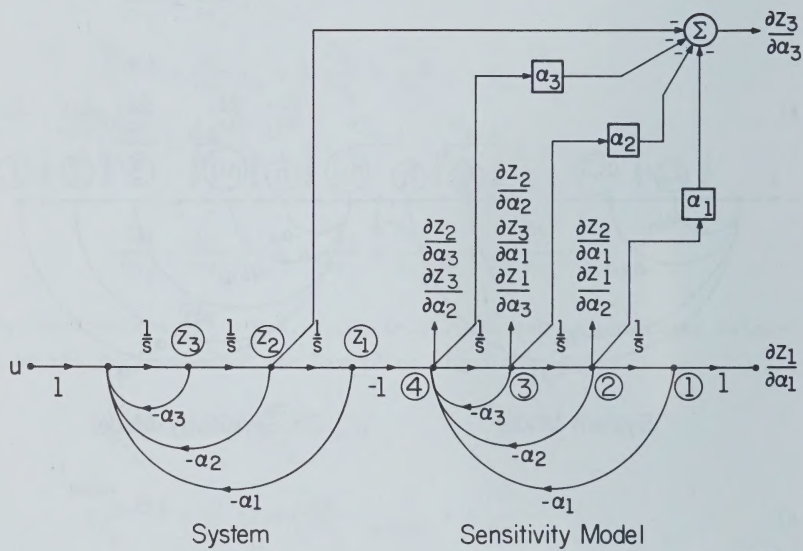
$$\begin{aligned}\frac{\partial \alpha_{n+1}}{\partial v_i} &= 0 & \frac{\partial S_{n+1}}{\partial v_i} &= 0 \\ \frac{\partial \alpha_{n-j+1}}{\partial v_i} &= \left(-\frac{1}{j}\right) \operatorname{tr} \left(\frac{\partial A}{\partial v_i} S_{n-j+2} + A \frac{\partial S_{n-j+2}}{\partial v_i} \right) \\ \frac{\partial S_{n-j+1}}{\partial v_i} &= \frac{\partial \alpha_{n-j+1}}{\partial v_i} I + \frac{\partial A}{\partial v_i} S_{n-j+2} + A \frac{\partial S_{n-j+2}}{\partial v_i}.\end{aligned}\quad (\text{A.4})$$

Thus, by using (A.1) through (A.4), $T(\underline{y})$, $\frac{\partial T}{\partial v_i}$, and $\frac{\partial \alpha_k}{\partial v_i}$, $k=1, \dots, n$, $i=1, \dots, r$ at any point $\underline{y}$ can be calculated assuming only the functional dependence of A and $\tilde{b}$ on $\underline{y}$ are known.



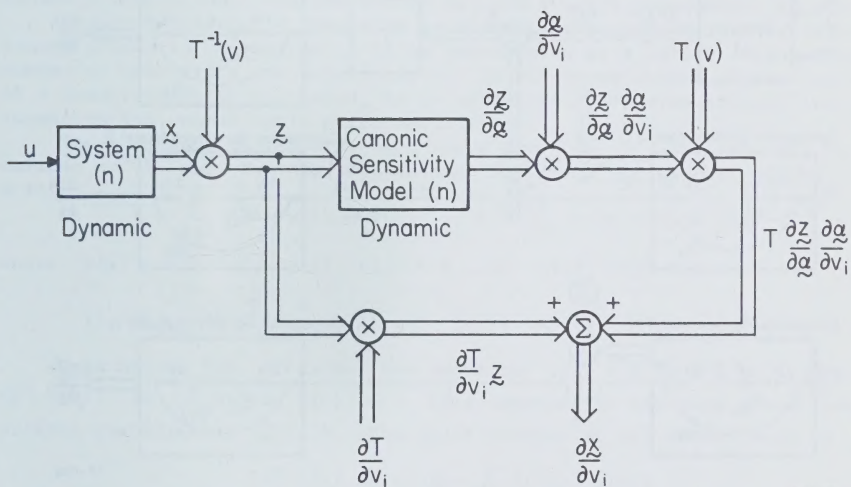
FP-1529

Figure 1. N^{th} Order Companion System and Sensitivity Model.



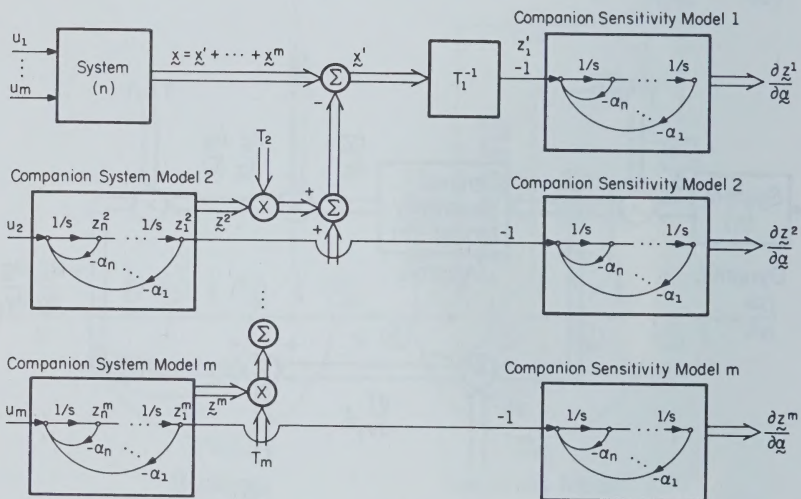
FP-1530

Figure 2. Third Order Example.



FP-1528

Figure 3. Block Diagram for Generating the Sensitivity Functions.



FR-1732

Figure 4. Generation of Sensitivity Functions for Normal System with m Inputs.

LEWIS MEIER
Stanford Research Institute
Menlo Park, California

ABSTRACT

This paper treats the application of tensor algebra to some system problems. After defining the necessary tensor concepts, the moment tensor and its dual the sensitivity tensor are discussed.

BASIC DEFINITIONS

Rudiments of Tensor Algebra

Tensor Product. Let X and Y be vector spaces with bases α_i and β_j ; then the tensor product $X \otimes Y$ is the vector space with basis $\alpha_i \beta_j$. Let $x \in X$ have components x^i and $y \in Y$ have components y^j ; then the tensor product $x \otimes y$ is the vector in $X \otimes Y$ with components $x^i y^j$.

Tensor Notation. A tensor is an element of a space formed by taking tensor products. Symbolically it is represented by a letter followed by a number of subscripts and superscripts. If all permutations of the indices of a tensor leave it unchanged, it is symmetric. Repeated indices imply summation and commas imply differentiation.

Inner Product of Tensors. Let inner products $(\cdot, \cdot)_x$ and $(\cdot, \cdot)_y$ be defined on X and Y . Any two tensors t and s in $X \otimes Y$ can be written

$$t = \sum_{i=1}^p a(i) x(i) \otimes y(i) \quad s = \sum_{j=1}^q b(j) u(j) \otimes v(j) \quad (1)$$

where $x(i), u(j) \in X, y(i), v(j) \in Y$, and $a(i), b(j) \in R$.

$$(t, s)_{X \otimes Y} \triangleq \sum_{i=1}^p \sum_{j=1}^q a(i) b(j) (x(i), u(j))_x (y(i), v(j))_y \quad (2)$$

Weak Direct Sum. Consider the sequence $t(i) \in X^i \triangleq X \otimes X \dots \otimes X$ ($X^0 \triangleq R$). If $\sum_{i=0}^{\infty} (t(i), t(i)) < \infty$ this sequence is the weak direct sum written symbolically $\bigoplus t(i)$. The inner product of two direct sums is

$$(\bigoplus t(i), \bigoplus s(i)) = \sum_{i=0}^{\infty} (t(i), s(i)) \quad (3)$$

Taylor's Series

Consider the scalar function f of the vector x^i . $x^* \triangleq 1 \oplus x^i \oplus \frac{1}{2} x^i \otimes x^i \oplus \frac{1}{3!} x^i \otimes x^i \otimes x^i \oplus \dots$ exists since $\sqrt{(x^*, x^*)} \leq e^{\sqrt{(x^i, x^i)}}$. Let $f'(x^i) = \bigoplus f^{,i_1 \dots i_j}(x^i)$. (If (f', f') is not finite, it may be made finite by replacing x^i with x^i/ϵ for sufficiently small ϵ .) The Taylor's series expansion of $f(x^i)$ about the origin is $(f'(0), x^*)$.

THE MOMENT AND SENSITIVITY TENSORS

The System

$$\dot{x}^i = f^i(x^i, t) + \sigma^i_j(x^i, t) w^j \quad (4)$$

where w is Gaussian white noise with covariance δ^i_j .

The Probability Density and its Properties

The Fokker-Planck Equation. The probability density $\hat{v}(x^i, t)$ that the system is in state x^i at time t obeys

$$\frac{\partial \hat{v}}{\partial t} = (f^i \hat{v})_{,i} - \frac{1}{2} (\sigma^i_j \sigma^j_k \hat{v})_{,i}^k \quad (5)$$

The Moment Tensor.

$$\hat{p}(t) \triangleq \int x^* v(x^i, t) dx^i = \hat{\otimes} \hat{p}^{i_1 \dots i_j(j)} \quad (6)$$

It is the weak direct sum of the moments of the probability density and obeys

$$\dot{\hat{p}} = \text{sym} \left\{ \hat{\otimes} \sum_{k=0}^j \binom{j}{k} \left[f^{i_{j+1}} \oplus \sigma^{i_{j+1}}_{\ell} \sigma^{\ell}_m \delta^{m i_{j+2}} \right]_{,i_1 \dots i_k} \hat{p}^{i_1 \dots i_j(j)} \right\} \quad (7)$$

where $\text{sym} \{ \}$ means that each term of the weak direct sum in its argument is made symmetric by averaging over all permutations.

The Return Function and its Properties

The Generalized Hamilton-Jacobi Equation. Let the performance of the system be

$$J = \int v(x^i, t_f) \hat{v}(x^i, t_f) dx^i \quad (8)$$

and let $v(x, t)$ be the expected value of J given the system is in state x^i at time t . Then $\int v(x^i, t) \hat{v}(x^i, t) dx$ is a constant; therefore

$$-\frac{\partial v}{\partial t} = v_{,i} f^i + \frac{1}{2} v_{,ik} \sigma^i_j \sigma^j_{\ell} \delta^{\ell k} \quad (9)$$

The Performance Tensor. Suppose $v(x^i, t) = (p(t), x^*)$. Then $(p(t), \hat{p}(t))$ is a constant and

$$-\dot{\hat{p}} = \text{sym} \left\{ \hat{\otimes} \sum_{k=0}^j \binom{j}{k} \left[f^m \oplus \frac{1}{j+1} \sigma^m_{\ell} \sigma^{\ell}_{i_j} \right]_{,i_1 \dots i_k} p^{i_{k+1} \dots i_{j+1}(j-k)} \delta^i_m \right\}^{i_{j+1}} \quad (10)$$

The Sensitivity Tensor. If some of the parameters of the system are unknown, then the difference between the actual parameters and nominal values of the parameters can be made additional state variables. The change in performance can be computed from Eq. (10); those portions of p corresponding to the parameters form the sensitivity tensor. In particular the first and second order sensitivities of Reference 1 are two terms in this tensor.

Optimization

Estimation. The conditional probability density of the state of the system, Eq. (4), obeys a generalization of the Fokker-Planck Equation, which may be converted to an equation for the conditional moment tensor in a manner similar to the above development. The conditional moment equation may in turn be used to develop approximate non-linear filters.

Control. If a controller structure (including an estimator) is included in the system and controller parameters made state variables, then the above theory can be used to compute performance as a function of the controller parameters. A gradient technique may be utilized to determine optimum controller parameters.

CONCLUSIONS

Two applications of tensor algebra to system theory have been presented. Tensor notation is tailor-made for easy computer programming; however, memory and speed limitation provide a limit to the number of terms in a weak direct sum that may be computed.

REFERENCE

1. Meier, "Performance Sensitivity of Control Systems to Parameter Uncertainties, Disturbances, and Measurement Noise," Proc. of the Fifth Annual Allerton Conference on Circuit and System Theory, University of Illinois, pp. 721-728 (October 4-6, 1967).

C. K. Sanathanan and D. C. Reddy
 Department of Information Engineering
 University of Illinois, Chicago Circle
 Chicago, Illinois

INTRODUCTION

Modal analysis techniques have been applied successfully to the analysis and design of distributed parameter feedback systems [1, 2, 3, 4, 6]. In particular, the work of Gould and Murray-Lasso [1] is quite significant in that their method using a finite number of space modes result in design procedures for the implementation of a feedback controller. However, their method is restricted to systems with homogeneous boundary conditions. This paper presents a different approach to the representation of a distributed parameter system. Using a finite (but sufficiently large) number of space modes, the more general and practical case of nonhomogeneous boundary conditions is treated. Such a representation is then used in the analysis and reduction of parameter sensitivity.

MODAL ANALYSIS

Gould and Murray-Lasso use the eigenfunctions of the system as the space modes. Such a choice of space modes makes the mathematical operations of the analysis conceptually easier. However, in practical problems, the determination of the eigenfunctions is cumbersome. Moreover, this approach is not suitable for systems with nonuniform boundary conditions.

In the present approach, the response of the system is expressed as a series consisting of a linear combination of a finite set of "suitable" space modes with time dependent coefficients. The convergence of such a series depends strongly upon the proper choice of the space modes. In many practical problems [3,4] it is found that the space modes which efficiently represent the response corresponding to the normal operating steady state conditions likewise do so for the transient conditions. This is based on the assumption that the spatial harmonic content of the transient response is not appreciably different from that of the steady state response. A final choice of the space modes must however be made only after a qualitative understanding of the transient behavior. Let the system be defined by the partial differential operator, $\mathcal{L}$.

$$\mathcal{L}\{y(x,t)\} = m(x,t) \quad (1)$$

Let,

$$y(x,t) = \sum_{i=1}^N \theta_i(t) \phi_i(x) \quad (2)$$

and

$$m(x,t) = \sum_{i=1}^N \mu_i(t) \phi_i(x) \quad (3)$$

where $y(x,t)$ is the response, $m(x,t)$ the actuating signal, $\mu_i(t)$ the given time dependent mixing coefficients, $\theta_i(t)$ the unknown mixing coefficients of the response, and $\phi_i(x)$ the space modes. The θ_i 's are determined by the following equations:

a) Boundary Conditions (n algebraic equations)

$$\sum_{i=1}^N \phi_i(x_k) \theta_i(t) = f_k(t) \quad k = 1, 2, \dots, n \quad (4)$$

where x_k is a boundary point and there are n such boundary points.

b) Equations Residual [5] (N-n differential equations)

$$\int_{\text{all } x} [\mathcal{L} \{ \sum_{i=1}^N \theta_i(t) \phi_i(x) \} - \sum_{i=1}^N \mu_i(t) \phi_i(x)] \phi_j(x) dx = 0 \quad (5)$$

$j = 1, 2, \dots, (N-n)$

Substituting equations (4) into equations (5), one obtains the standard form:^{*}

$$\dot{\theta} = A\theta + BU(t) \quad (6)$$

SENSITIVITY

Having thus reduced the representation of a distributed system to that of an equivalent lumped system, the techniques of sensitivity reduction as applied to lumped systems can now be extended to the distributed systems. The response of the system at any specific space point, x_0 , can be written as:

$$y(x_0, t) = \sum_{i=1}^N w_i \theta_i(t), \quad w_i \equiv \phi_i(x_0) \quad (7)$$

Sensitivity of this response to the variation of a parameter, α , is:

$$\zeta(t) = \frac{\partial y(x_0, t)}{\partial \alpha} = \sum_{i=1}^N w_i \frac{\partial \theta_i(t)}{\partial \alpha} = \sum_{i=1}^N w_i z_i(t) \quad (8)$$

where the w_i 's can be considered as weighting functions defining the space point, and the z_i 's are the sensitivity functions. It can be shown that the "norm" of $\zeta(t)$ can be reduced by means of a feedback matrix F , which modifies A to $(A+F)$ such that the "norm" of $(A+F)$ is less than the "norm" of A . This result can be shown to be a generalization of that due to Porter [6]. It is further noted that the sensitivity reduction holds for all space points.

BIBLIOGRAPHY

- [1] L. A. Gould and M. A. Murray-Lasso, IEEE transactions, Vol.AC-11, No.4, 729-737, (1966).
- [2] W. A. Porter, Int. J. Control, Vol. 5, No. 5, 393-412, (1967)
- [3] C. K. Sanathanan et. al., Nuclear Science and Engineering Vol.23, 119-136, (1965).
- [4] S. Kaplan, et. al., Nuclear Science and Engineering, Vol. 18, 163-176, (1964).
- [5] S. H. Crandall, "Engineering Analysis," Mc Graw-Hill, New York, (1956).
- [6] W. A. Porter, Int. J. Control, Vol. 5, No. 1, 1-9, (1967).

^{*}The matrix A in eq. (6) is not, in general, diagonal. The vector, $U(t)$ is constituted by the time dependent coefficients $\mu_i(t)$ of the forcing function, $m(x, t)$ and the boundary condition $f_k(t)$ applied at each boundary point x_k . B is a constant matrix.

LOW-SENSITIVITY DESIGN OF THE LINEAR MINIMAX OUTPUT REGULATOR PROBLEM

RONALD D. BONNELL
University of Kentucky
Lexington, Kentucky

ROGER P. WEBB and
STEVEN L. DICKERSON
Georgia Institute of Technology
Atlanta, Georgia

ABSTRACT

This paper considers the problem of designing a minimax output regulator with constraints on the sensitivity of the state trajectory. A linear time-varying multi-input-multi-output dynamic system is taken to be the model of the plant with the disturbances entering into the states of the plant in an additive fashion. The plant is assumed to be completely state observable. The control vector and the disturbance vector are unconstrained.

The sensitivity vector employed is defined by the partial derivative of the plant state vector with respect to a plant parameter, evaluated at the nominal value of the plant parameter. The dynamic behavior of the sensitivity vector is described by a linear, time-varying differential equation with an initial sensitivity vector of zero. The sensitivity equations are derived under the assumption that the plant parameters remain constant throughout a given plant run. Although the formulation presented is for a single plant parameter, it can easily be extended to include more than one parameter.

The performance index used for the differential game is a general quadratic performance index which includes two quadratic forms for the terminal cost of the plant output and the sensitivity vector, and the integral over a finite time interval of four quadratic forms for the cost of the plant output, sensitivity, control and disturbance. By varying the weighting matrices in the performance index it is possible to examine the relative effect on the design of the system of each of these factors.

By considering an augmented dynamic system, the low-sensitivity design problem can be reformulated as a linear minimax output regulator problem where the augmented state vector consists of the plant state vector and the sensitivity vector.

The basic problem is to determine the "best" control and the "worst" disturbance which "minimaxes" the performance index. The performance index satisfies the so-called minimax condition as well as the saddle point condition. The game is one of perfect information in that the designers of both the control and the disturbance know all the information given in the preceding formulation.

The existence and uniqueness of the solution of the game are specified in terms of the relative controllability of the augmented system.

The "best" control is determined to be negative feedback of the augmented state while the "worst" disturbance is positive feedback of the augmented state. Two examples are presented to demonstrate the effect of varying the various weighting matrices in the performance index. The results of this investigation are useful in studying low-sensitivity designs of the output regulator for the "worst-case" disturbance situation.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant NSF-GE-4743.2.

THOMAS G. MANCUSO and SAMUEL C. LEE
Department of Electrical Engineering, New York University,
Bronx, New York 10453

ABSTRACT

In the process of digital filtering with floating-point arithmetic, three degradations exist; (1) quantization of input signal, (2) quantization of system parameters and (3) roundoff errors in computation. A bound on the fractional error in the magnitude of the response for all filter structures, in terms of the above three errors is presented. Based on the error criterion, for a given transfer function, the "best" structure among all possible realizations can be determined. A simple example illustrates the selection of the best structure. A comparison is made with results from digital filters programmed on a digital computer. Furthermore, for a given tolerance specification, the determination of a "minimum" word-length filter is also discussed in this paper.

I. INTRODUCTION

A digital filter is a sequence of operations which act on input sequence of numbers to give an output sequence of numbers. The digital filter simulates a linear difference equation. An L th order difference equation can be written as

$$Y_n = \sum_{i=0}^M b_i X_{n-i} - \sum_{i=1}^L a_i Y_{n-i} \quad (1)$$

where $\{Y_n\}$ is the output sequence and $\{X_n\}$ is the input sequence. Implementation of (1) using digital elements (e.g. shift registers, adders, multipliers, gates, or general purpose computers) is shown in Fig. 1a and is referred to as the direct method. Higher order difference equations may be formed using combinations of first and second order equations. An example of such is

$$\begin{aligned} Y_{m,n} &= \sum_{i=0}^{M_1} b_{1,i} X_{n-i} - \sum_{i=1}^{N_1} a_{1,i} Y_{m,n-i} \\ Y_{m-1,n} &= \sum_{i=0}^{M_2} b_{2,i} Y_{m,n-i} - \sum_{i=1}^{N_2} a_{2,i} Y_{m-1,n-i} \\ &\vdots \\ Y_n &= \sum_{i=0}^M b_{m,i} Y_{1,n-i} - \sum_{i=1}^N a_{m,i} Y_{n-i} \end{aligned} \quad (2)$$

Implementation of (2) (Fig. 1b) is called the cascade method. Another example is

$$\begin{aligned} Y_{1,n} &= \sum_{i=0}^{M_1} b_{1,i} X_{n-i} - \sum_{i=1}^{N_1} a_{1,i} Y_{1,n-i} \\ &\vdots \\ Y_{m,n} &= \sum_{i=0}^M b_{m,i} X_{n-i} - \sum_{i=1}^N a_{m,i} Y_{m,n-i} \\ Y_n &= \sum_{i=1}^m Y_{i,n} \end{aligned} \quad (3)$$

the implementation of which is the parallel method (Fig. 1c). Reference is made to a survey paper¹ which goes into considerable detail on the methods for determining the coefficients a_i and b_i of the difference equations.

The z transform of equation (1) is

$$Y(z) = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} X(z) = H(z) X(z) \quad (4)$$

where $H(z)$ is the system transfer function. Similarly for (2)

$$\begin{aligned} Y_m(z) &= \frac{\sum_{i=0}^{M_1} b_{1i} z^{-i}}{1 + \sum_{i=1}^{N_1} a_{1i} z^{-i}} X(z) = [H_1(z)] X(z) \\ Y_{m-1}(z) &= \frac{\sum_{i=0}^{M_2} b_{2i} z^{-i}}{1 + \sum_{i=1}^{N_2} a_{2i} z^{-i}} Y_m(z) = [H_2(z)] H_1(z) X(z) \\ &\vdots \\ Y(z) &= \frac{\sum_{i=0}^{M_m} b_{mi} z^{-i}}{1 + \sum_{i=1}^{N_m} a_{mi} z^{-i}} Y_1(z) = [H_m(z)] \dots H_2(z) H_1(z) X(z) \end{aligned} \quad (5)$$

and for (3)

$$\begin{aligned} Y_1(z) &= \frac{\sum_{i=0}^{M_1} b_{1i} z^{-i}}{1 + \sum_{i=1}^{N_1} a_{1i} z^{-i}} X(z) = [H_1(z)] X(z) \\ &\vdots \\ Y_m(z) &= \frac{\sum_{i=0}^{M_m} b_{mi} z^{-i}}{1 + \sum_{i=1}^{N_m} a_{mi} z^{-i}} X(z) = [H_m(z)] X(z) \\ Y(z) &= [H_1(z) + \dots + H_m(z)] X(z) = H(z) X(z) \end{aligned} \quad (6)$$

The magnitude of the frequency response, $|H(e^{j\omega T})|$, is of primary interest in this paper, and is interpreted in the same manner as in continuous systems.¹⁻²

There are three degradations in the process of digital filtering. These are: (1) quantization of the input signal, (2) quantization of system parameters and (3) roundoff errors in computation. Floating point arithmetic is considered since this mode of arithmetic gives a wider range of values as compared to fixed point. An expression will be derived for the fractional error and in the magnitude function for each of the three configurations and will be shown to be a function of these errors. One can then choose the form with the least error for a given transfer function. The fractional error in the magnitude function is

$$\max_{\omega} \left| \frac{\Delta |H(\omega)|}{|H(\omega)|} \right| < \sigma = \sigma_r + \sigma_i + \sigma_q \quad (7)$$

where σ_i is the upper bound on the fractional error due to quantization of the input, σ_q is the upper bound on the fractional error due to the quantization of the desired coefficients and σ_r is the upper bound on the fractional error due to roundoff errors in the arithmetic. The expression for σ_r and σ_i were derived by Sandberg³ which is the upper-bound on the asymptotic output error to signal ratio for a mantissa of t bits. The values of σ_i and σ_r require knowledge of the magnitudes of the numerator and denominator functions of $H(z)$, and the order in which the arithmetic operations are carried out. The third term, σ_q , represents the upper bound on the fractional error due to the coefficient quantization in the decimal to binary conversion and is obtained from the multiparameter sensitivity of $H(z)$ associated with each of the coefficients.

II. PARAMETER QUANTIZATION ERROR AND MULTIPARAMETER SENSITIVITY

A bound must be established which describes the maximum error in the magnitude of the response due to changes in the parameters. A useful technique for setting a bound on this error is the use of multiparameter sensitivity. The parameter quantization error is dependent on the method by which the original coefficients are introduced into the system (e.g. truncating or rounding to a given number of decimal digits) and how the decimal to binary conversion is accomplished. Both of these depend on t , the number of bits used in the internal representation of the coefficients. For an example, for a 2^4 bit machine the maximum resolution of the mantissa for this machine is 2^{-24} or 6×10^{-8} . Thus 6×10^{-8} is the maximum magnitude of the error in the mantissa which means that 7 digits can be represented with an error of ± 6 in the eighth decimal place. This criterion is applicable to any size machine. This error is inherent in any decimal to binary conversion. Also there will be an error if the original coefficient has to be represented by any other number. The largest fractional error is found to be

$$\delta_{\max} = \left[\frac{|\Delta a_i|}{|a_i \min|} \right] \quad (8)$$

where Δa_i is the maximum fractional error in the coefficients and $a_i \min$ is the smallest coefficient. In order to establish the relationship between the coefficient fractional error and the fractional error in the magnitude of the response, the multiparameter sensitivity of the transfer function due to coefficients is used. Single parameter sensitivity is defined as⁴

$$S_{a_i} \left| \frac{\partial |H(e^{j\omega T})|}{\partial a_i} \right| = \operatorname{Re} \left[\frac{\partial \ln H(e^{j\omega T})}{\partial \ln a_i} \right] \quad (9)$$

such that

$$\frac{\Delta |H(e^{j\omega T})|}{|H(e^{j\omega T})|} \leq (\max_{\omega} |S_{a_i}| |H(e^{j\omega T})|) \cdot \delta_{\max} \quad (10)$$

For the multiparameter case, the fractional error in the magnitude will be the sum of the magnitude of the individual sensitivities multiplied by the maximum coefficient fractional error

$$\frac{\Delta |H(e^{j\omega T})|}{|H(e^{j\omega T})|} \leq (\max_{\omega} \sum_i |\operatorname{Re} S_{a_i}| |H(e^{j\omega T})|) \cdot \delta_{\max} \quad (11)$$

The bound on the fractional error due to the coefficient quantization will be

$$\sigma_q = \left| \bar{S} |H(\omega)| \right| \cdot \delta_{\max} \quad (12)$$

where

$$\left| \bar{S} |H(\omega)| \right| = (\max_{\omega} \sum_i |\operatorname{Re} S_{a_i}| |H(e^{j\omega T})|)$$

Note that the value σ_q will be zero when all the a_i are represented exactly as machine numbers. Generally the calculation of multiparameter sensitivity requires the use of a computer.

III. DETERMINATION OF ROUND-OFF AND INPUT QUANTIZATION ERRORS

The effects of round-off and input quantization were discussed in a recent paper by Sandberg.³ It has been shown that for round-off there is a quantity C which sets the bound on the asymptotic output error to signal ratio. The value of σ_r is approximately C , for small output error

$$\sigma_r \approx C \leq 2^{-t} \max_{\omega} |D(e^{j\omega T})|^{-1} \left\{ \sum_k |a_k| \alpha_k + \frac{\sum_k |b_k| \beta_k \cdot \max_{\omega} |D(e^{j\omega T})/N(e^{j\omega T})| + \max_{\omega} |N(e^{j\omega T})|^{-1} 2^{-t} \sum_k |a_k| \alpha_k}{1 - 2^{-t} \sum_k |b_k| \beta_k \max_{\omega} |N(e^{j\omega T})|^{-1}} \right\} \quad (13)$$

such that

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_k b_k z^{-k}}{\sum_k a_k z^{-k}} \quad (14)$$

and

$$\alpha_k = 1.06 \phi_{\alpha}(k)$$

$$\beta_k = 1.06 \phi_{\beta}(k)$$

where $\phi_{\alpha}(k)$ and $\phi_{\beta}(k)$ are positive-integer valued functions, t is the number of bits in the mantissa of the floating-point machine number. For further information regarding these equations see reference 3.

There is also a possibility that the input signal can be quantized (i.e. decimal to binary conversion). Sandberg³ has also given some results for the determination of the maximum fractional error in the magnitude of the response due to input quantization. The value of σ_i is

$$\sigma_i \leq 2^{-t} \sum_k |b_k| \max_{\omega} |D(e^{j\omega T})|^{-1} \left\{ \frac{\max_{\omega} |D(e^{j\omega T})/N(e^{j\omega T})| + \max_{\omega} |N(e^{j\omega T})|^{-1} 2^{-t} \sum_k |a_k| \alpha_k}{1 - 2^{-t} \sum_k |b_k| \beta_k \max_{\omega} |N(e^{j\omega T})|^{-1}} \right\} \quad (15)$$

IV. AN UPPER-BOUND ON THE FRACTIONAL ERROR OF THE MAGNITUDE

For the direct method, the upper bound on the fractional error in the magnitude is the sum of the fractional errors

$$\max_{\omega} \frac{\Delta |H(\omega)|}{|H(\omega)|} \leq \sigma = \sigma_r + \sigma_i + \sigma_q \quad (16)$$

From Equation (6), $H(z)$ for the parallel method is composed of a sum of first and/or second order transfer functions

$$H(z) = H_1(z) + H_2(z) + \dots \quad (17)$$

In order to find $\max_{\omega} \frac{\Delta |H(\omega)|}{|H(\omega)|}$ for the parallel method, the procedures used in section III will be applied to each $H_i(z)$. To find $\max_{\omega} \frac{\Delta |H(\omega)|}{|H(\omega)|}$, let

$$|H(\omega)| + \Delta |H(\omega)| = (|H_1(\omega)| + \Delta |H_1(\omega)|) + (|H_2(\omega)| + \Delta |H_2(\omega)|) + \dots \quad (18)$$

then

$$\Delta |H(\omega)| = \Delta |H_1(\omega)| + \Delta |H_2(\omega)| + \dots \quad (19)$$

from which

$$\frac{\Delta |H(\omega)|}{|H(\omega)|} = \frac{\Delta |H_1(\omega)| + \Delta |H_2(\omega)| + \dots}{|H_1(\omega)| + |H_2(\omega)| + \dots} = \frac{\Delta |H_1(\omega)|}{|H_1(\omega)| + |H_2(\omega)| + \dots} + \frac{\Delta |H_2(\omega)|}{|H_1(\omega)| + |H_2(\omega)| + \dots} + \dots \quad (20)$$

Using the triangular inequality,

$$\begin{aligned} \left| \frac{\Delta |H(\omega)|}{|H(\omega)|} \right| &\leq \left| \frac{\Delta |H_1(\omega)|}{|H_1(\omega)| + |H_2(\omega)| + \dots} \right| + \left| \frac{\Delta |H_2(\omega)|}{|H_1(\omega)| + |H_2(\omega)| + \dots} \right| + \dots \leq \left| \frac{\Delta |H_1(\omega)|}{|H_1(\omega)| + |H_2(\omega)|} \right| + \\ &\quad \left| \frac{\Delta |H_2(\omega)|}{|H_1(\omega)| + |H_2(\omega)|} \right| + \dots \\ \left| \frac{\Delta |H(\omega)|}{|H(\omega)|} \right| &\leq \left| \frac{\Delta |H_1(\omega)|}{|H_1(\omega)|} \right| + \left| \frac{\Delta |H_2(\omega)|}{|H_2(\omega)|} \right| + \dots \quad (21) \end{aligned}$$

The overall fractional error is found by summing the individual fractional errors

$$\max_{\omega} \left| \frac{\Delta |H(\omega)|}{|H(\omega)|} \right| \leq \sigma = \sum_k \sigma_{r_k} + \sum_k \sigma_{i_k} + \sigma_q \quad *(22)$$

For the cascade method from equation (5), the transfer function $H(z)$ is composed of products of first and/or second order transfer functions of the form

$$H(z) = H_1(z) \cdot H_2(z) \cdot \dots \quad (23)$$

Then

$$(|H(\omega)| + \Delta |H(\omega)|) = (|H_1(\omega)| + \Delta |H_1(\omega)|)(|H_2(\omega)| + \Delta |H_2(\omega)|) \dots \quad (24)$$

*Since we are considering the fractional change on the overall $|H(e^{j\omega T})|$ due to each coefficient, $\sum \sigma_{q_k} \neq \sigma_q$.

or

$$\begin{aligned} \Delta |H(\omega)| &= \Delta |H_1(\omega)| \cdot |H_2(\omega)| \cdot |H_3(\omega)| \cdots + |H_1(\omega)| \cdot \Delta |H_2(\omega)| \cdot |H_3(\omega)| \cdots + \dots \\ &+ \Delta |H_1(\omega)| \cdot \Delta |H_2(\omega)| \cdot |H_3(\omega)| \cdots + \dots \end{aligned}$$

from which

$$\frac{\Delta |H(\omega)|}{|H(\omega)|} \approx \frac{\Delta |H_1(\omega)|}{|H_1(\omega)|} + \frac{\Delta |H_2(\omega)|}{|H_2(\omega)|} + \dots \quad (25)$$

for small $\Delta |H_i(\omega)|$. Using the triangular inequality on (25)

$$\left| \frac{\Delta |H(\omega)|}{|H(\omega)|} \right| < \left| \frac{\Delta |H_1(\omega)|}{|H_1(\omega)|} \right| + \left| \frac{\Delta |H_2(\omega)|}{|H_2(\omega)|} \right| + \dots \quad (26)$$

Notice that (26) is the same as (21). Thus the bound on the fractional error for the cascade method has the same expression as (22).

In general, a given transfer function can be decomposed into combinations of products and sums of first and/or second order transfer functions, the realization being combinations of cascade and parallel method filters. A typical structure is shown in Fig. 1d. The overall bound on the fractional error is found by first finding the individual fractional errors and then repeatedly using (22). Therefore the general formula for determining the bound on the fractional error for the direct, cascade, or parallel method or any combinations of these methods is

$$\max_{\omega} \left| \frac{\Delta |H(\omega)|}{|H(\omega)|} \right| \leq \sigma = \sum_{\substack{\text{all indi-} \\ \text{vidual sub-} \\ \text{realizations} \\ \text{of } H_k(z)}} \sigma_{r_k} + \sum_{\substack{\text{all indi-} \\ \text{vidual sub-} \\ \text{realizations} \\ \text{of } H_k(z)}} \sigma_{i_k} + \sigma_q \quad (27)$$

V. DETERMINATION OF THE BEST STRUCTURES

From the previous discussions and (27), for a given transfer function the best structure may be found by applying (27) to all possible structures and choosing the one with the least fractional error for a specified word-length. This is illustrated by the following example.

Example

Suppose that the given transfer function is

$$H(z) = \frac{0.1760288 z^{-1} + 0.1047878 z^{-2}}{1 - 1.506018 z^{-1} + 0.9347095 z^{-2} - 0.2078795 z^{-3}} \quad (28)$$

$$= \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \quad (28)$$

which is a three pole lowpass filter with a cutoff at approximately $\frac{1}{4}$ of the sampling frequency.

Direct Method

Using (13) and reference 3, p. 1779, $\phi_\alpha(1)=4$, $\phi_\alpha(2)=4$, $\phi_\alpha(3)=3$, $\phi_\beta(0)=0$, $\phi_\beta(1)=4$ and $\phi_\beta(2)=3$. The α_i and β_i associated with a_i and b_i , respectively are found to be $\alpha_1=4.24$, $\alpha_2=4.24$, $\alpha_3=3.18$, $\beta_0=0$, $\beta_1=4.24$ and $\beta_2=3.18$. Also $\max_{\omega} |D(e^{j\omega T})^{-1}|$, $\max_{\omega} |N(e^{j\omega T})^{-1}|$ and $\max_{\omega} |D(e^{j\omega T})/N(e^{j\omega T})|$ are found to be 4.65, 14.2 and 52 respectively. The terms

$(\max_{\omega} |N(e^{j\omega T})^{-1}| 2^{-t} \sum_k |a_k| \alpha_k)$ and $(2^{-t} \sum_k |b_k| \beta_k \max_{\omega} |N(e^{j\omega T})^{-1}|)$ are approxi-

mately zero for large t . Computing (13), with $t = 24$, yields $\sigma_1 = 0.407 \times 10^{-5}$. The plot of $\sum_i |\operatorname{Re} S_{a_i}^H(\omega)|$ is shown in Fig. 2. It is seen that the maximum

value is 12.99 at $\omega T = 0$. Since the smallest coefficient a_i min is 0.1047878, and $\Delta a_i = 2^{-t} = 6 \times 10^{-8}$ for $t = 24$. Thus σ_q is calculated to be 0.745×10^{-5} . Using (27), the upper bound on the fractional error of the magnitude, σ is 0.3012×10^{-4} .

Parallel Method

The transfer function (28) is decomposed into partial fractions such that

$$H(z) = \frac{1}{1 - 0.4559380z^{-1}} + \frac{-1 + 0.7701709z^{-1}}{1 - 1.050079z^{-1} + 0.4559380z^{-2}} \quad (29)$$

$$= H_1(z) + H_2(z)$$

Proceeding as in the direct method, for $H_1(z)$, δ_{r_1} is 0.267×10^{-6} , $\delta_{i_1} = 0.161 \times 10^{-6}$. Similarly for $H_2(z)$, $\delta_{r_2} = 0.1595 \times 10^{-5}$ and $\delta_{i_2} = 0.467 \times 10^{-6}$.

The value of δ is 0.274×10^{-4} where $\max_{\omega} \sum_i |\operatorname{Re} S_{a_i}^H(\omega)| \approx 48$. Using the general formula (27), δ is found to be 0.299×10^{-4} , for a 24 bit machine.

Cascade Method

The transfer function for the cascade method is the factored form of (28)

$$H(z) = \left(\frac{0.1760289z^{-1}}{1 - 0.4559380z^{-1}} \right) \left(\frac{1 + 0.5952875z^{-1}}{1 - 1.050079z^{-1} + 0.4559380z^{-2}} \right) = H_1(z) \cdot H_2(z) \quad (30)$$

Once again, the method used in the direct method is carried out for $H_1(z)$ and $H_2(z)$. For $H_1(z)$, $\delta_{r_1} = 0.446 \times 10^{-6}$ and for $H_2(z)$, $\delta_{r_2} = 0.335 \times 10^{-5}$, $\delta_{i_2} = 0.141 \times 10^{-5}$. The value of δ_q is 0.280×10^{-5} . The general formula (27), gives $\delta = 0.813 \times 10^{-5}$.

Note that since this is a 3rd order system, only these three methods are possible, otherwise the coefficients would not be real. A summary of the values for this example is given in Table 1. Of the three structures used for this transfer function, the cascade has the smallest fractional error in the magnitude.

Experimental Verification

Verification of the results obtained above was accomplished in the following manner. Each of the three filter methods was programmed on the IBM 360 machine. The filters were run twice, once for single precision variables, $t = 24$, and the other for double precision, $t = 56$ bits. The latter was used as a standard value. A unit sinusoid, for various values of ωT in the interval 0 to η , was used as the excitation. The RMS value of

a sequence $\{e_n\}$, E_{RMS} , is defined by

$$E_{\text{RMS}} \triangleq \left(\frac{1}{k+1} \sum_{n=0}^k |e_n|^2 \right)^{\frac{1}{2}} \quad (31)$$

The output signal was examined and the RMS value of the output sequence was obtained for both the single and double precision cases. The relative error, which is the difference between the single and double precision results divided by the double precision result, was then calculated and in fact is the value of $\frac{\Delta |H(\omega)|}{|H(\omega)|}$. The results of the predicted and experimental values

for the three filters are tabulated and compared in Table 1. In this example, the cascade method filter is found to be the best structure, since it has the smallest error.

VI. DIGITAL FILTER DESIGN FOR MINIMUM WORD-LENGTH

One of the problems in digital filter desing is to implement a given transfer function with a structure which gives the least detriment to the desired output, with the smallest possible word-length. The shortest word-length provides highest operating speed while reducing the cost of the over-all filter. For a given tolerance on the magnitude of the response, an iterative process is used to find the structure which meets this tolerance while maintaining minimum word-length. It must be remembered that the stability of the system must be considered, since quantization of system parameters might cause the poles of $H(z)$ to move outside the unit circle. A lower bound on the minimum number of bits required to insure stability, by finding pole sensitivity with respect to the coefficient changes has been shown in the literature.⁵ A computer flow chart for finding the structure with minimum word-length requirements for a given tolerance specification is given in Fig. 3.

VII. SUMMARY

A general formula for finding the upper bound on the fractional error of the magnitude of any digital filter structure has been presented. Using this formula the "best" structure may be found for realizing a given transfer function, and an example has been given to illustrate this. The result obtained in this example was experimentally verified. A computer flow chart for finding the structure with minimum word-length requirements for a given tolerance specification was presented.

REFERENCES

1. C.M. Rader and B. Gold, "Digital Filter Design Techniques in the Frequency Domain," Proc. IEEE, vol. 55, no. 2, February 1967.
2. J.R. Ragazzini and G.F. Franklin, Sampled-Data Control Systems. New York: McGraw-Hill, 1958.
3. I.W. Sandberg, "Floating-Point-Roundoff Accumulation in Digital-Filter Realizations," B.S.T.J., vol. XLVI, no. 8, October 1967.
4. J.B. Cruz, Jr., and W.R. Perkins, "A New Approach to the Sensitivity Problem in Multivariable Feedback System Design," IEEE Trans. on Auto. Control, vol. AC-9, July 1964.
5. J.F. Kaiser, "Some Practical Considerations in the Realization of Linear Digital Filters," Proc. 3rd Ann. Allerton Conf. on Circuit and System Theory, October 1965.

TABLE 1

		<u>DIRECT</u>	<u>PARALLEL</u>	<u>CASCADE</u>
σ_r		0.1860×10^{-4}	0.186×10^{-5}	0.380×10^{-5}
σ_i		0.407×10^{-5}	0.063×10^{-5}	0.153×10^{-5}
σ_q		0.745×10^{-5}	0.274×10^{-4}	0.280×10^{-5}
σ	predicted	0.3012×10^{-4}	0.299×10^{-4}	0.813×10^{-5}
$\frac{\Delta H(\omega) }{ H(\omega) }$	measured	0.473×10^{-5}	0.131×10^{-4}	0.309×10^{-5}

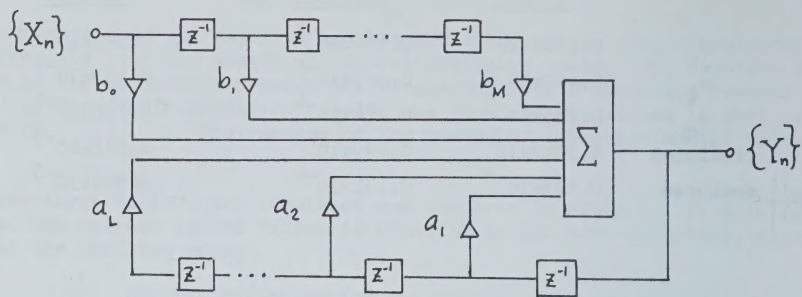


Figure 1a Direct Method

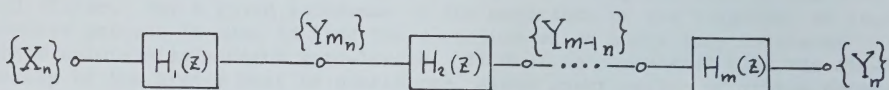


Figure 1b Cascade Method

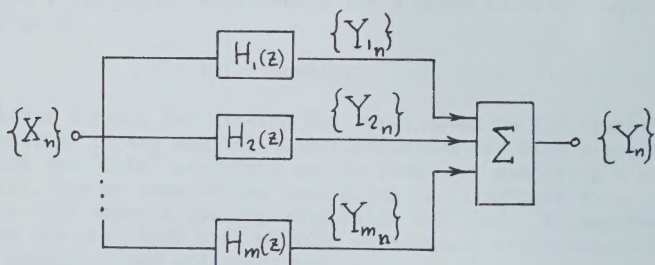


Figure 1c Parallel Method

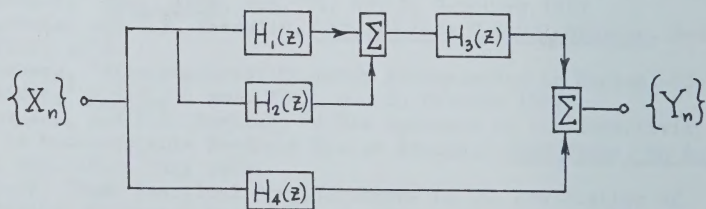


Figure 1d A typical structure for General Method

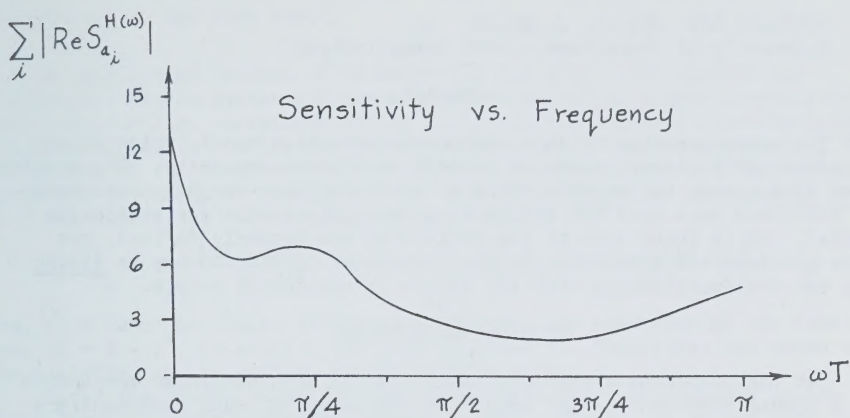


Figure 2

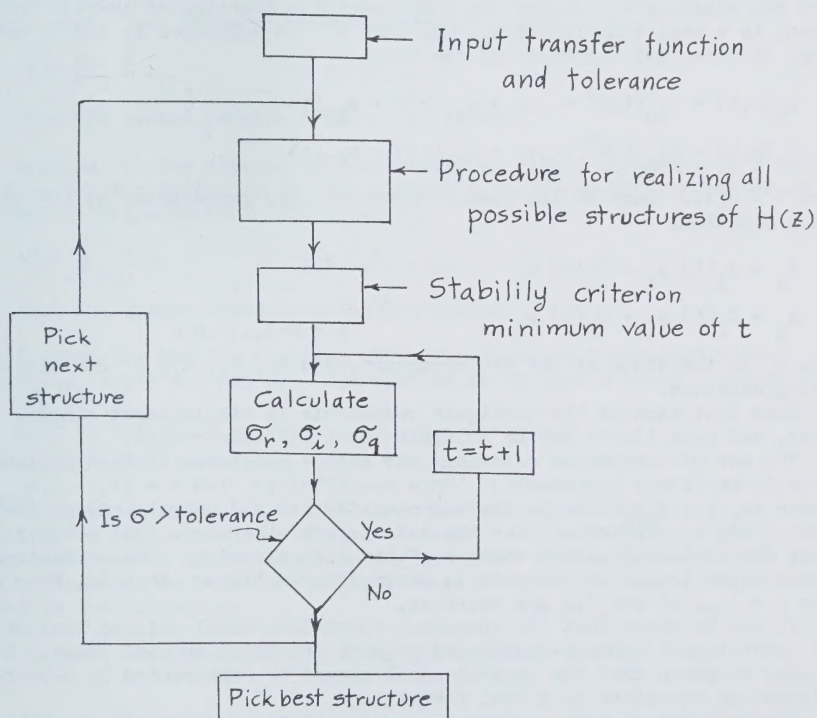


Figure 3

Flow chart

RUEY-WEN LIU AND R. J. LEAKE

University of Notre Dame, Notre Dame, Indiana

ABSTRACT

The system considered is a component-connection model, which allows any number of nonlinear elements as well as linear components. The problem is to find bounds on the variations of nonlinear time-varying characteristic functions such that the system is dynamically stable (or statically stable). It is found that if the variations are properly defined, the above problems are equivalent to the problem of finding bounds on linear time varying functions so that the system is statically stable.

PROBLEM AND RESULTS

Let the component-connection model consist of p nonlinear components and q linear components. The input and the output of each component are denoted by v_i and x_i . Let the input and the output of each of the nonlinear components be scalars related by the equations

$$L_i(D, t) f_i(x_i, t) = M_i(D, t) v_i \quad i = 1, 2, \dots, p$$

where the function f_i , called the characteristic function of the i th component, is a real function on R_2 , and each of the operators L_i and M_i are linear differential operators of the form

$$L_i(D, t) = a_{in}(t)D^n + \dots + a_{i1}(t)D + a_{i0}(t)$$

$$M_i(D, t) = b_{im}(t)D^m + \dots + b_{i1}(t)D + b_{i0}(t)$$

where $D = d/dt$. Each of the linear components are represented by the usual vector equations

$$\dot{s}_i = A_i(t) s_i + B_i(t) v_i$$

$$x_i = C_i(t) s_i + D_i(t) v_i \quad i = p+1, \dots, p+q$$

where s_i is the state of the i th component, and A_i , B_i , C_i , D_i are time-varying matrices.

Note that each of the nonlinear components is single-input single-output, and each linear may be multiple-input multiple-output.

The set of connection equations may not be nonlinear (unless included as static nonlinear components). More specifically, let $v = (v_1, \dots, v_{p+q})$ and $x = (x_1, \dots, x_{p+q})$ denote the component inputs and outputs respectively, where v_i and x_i are vectors for $i > p$ and scalars otherwise. Let u and y denote the input and output vectors of the entire system. The connection between these inputs and outputs is assumed to be linear, that is, $Pr = Qu$, where $r = (y, x, v)$ and P, Q are matrices.

It can be shown that the component-connection model defined includes most conventional control system and network models as special cases. It can also be shown that the general model cannot be represented by n -vector differential equations in normal form.

The problems at hand concern the system static and dynamic stability. The system is said to be statically stable iff when $u = 0$ every output approaches zero, $\lim_{t \rightarrow \infty} \|y(t)\| = 0$ and further $y = 0$ is a response.

The system is said to be dynamically stable, iff for any input u , the outputs are pairwise convergent, that is, iff when $y_1(t)$ and $y_2(t)$ are outputs due to the same input,

$$\lim_{t \rightarrow \infty} \|y_1(t) - y_2(t)\| = 0.$$

A sensitivity problem is to determine how well a dynamically (or statically) stable system can sustain variations of nonlinear components: more specifically, variations of the nonlinear time-varying characteristic functions $f_i(x, t)$. Unlike the variation of constant system parameters, the variations of functions can be made in many different ways. It is necessary to define the norms of functions. Let the static norm and the dynamic norm of a function $f(x, t)$ be defined respectively by

$$\|f\|_S = \sup_{x, t} \left| \frac{f(x, t)}{x} \right|, \quad \|f\|_D = \sup_{x, t} \left| \frac{\partial f(x, t)}{\partial x} \right|$$

Now, if a function $f^*(x, t)$ is varied to $f(x, t)$, the variation of the function, $\delta f = f - f^*$, is still a function of x and t . Therefore, the norms of the variation can be defined accordingly as above.

Since the static stability and dynamic stability depend only on the final function $f(x, t)$, not the initial function $f^*(x, t)$, we assume that the characteristic functions are initially linear, that is, $f_i^*(x, t) = k_i x_i$, where the k_i 's are constants used only for reference.

Problem A: The problem of sensitivity of dynamic stability of a nonlinear time-varying system is to find a bound $c = (c_1, c_2, \dots, c_p)$ on the dynamic variations of initial characteristic functions

$$\|\delta f_i\|_D \leq c_i, \quad i = 1, 2, \dots, p$$

such that the system remains dynamically stable.

Problem B: The problem of sensitivity of static stability of a nonlinear time-varying system is to find a bound c on the static variations of the initial characteristic functions f_i^* ,

$$\|\delta f_i\|_S \leq c_i, \quad i = 1, 2, \dots, p$$

such that the system remains statically stable.

Problem C: Let the characteristic functions be linear and time-varying, $f_i(x_i, t) = g_i(t) x_i$. The problem is to find a bound c such that if

$$k_i + c_i \geq g_i(t) \geq k_i - c_i, \quad i = 1, 2, \dots, p$$

then the linear time-varying system is statically stable.

It is well known that a problem of infinitesimal sensitivity of local stability of nonlinear time-varying systems can be reduced to a problem of local stability of linear time-varying systems. The main result here is that it can also be done for in-large sensitivity of global stability as stated in the following

Theorem: The problems (A, B, C) are equivalent.

REFERENCE

R. Liu, CONVERGENT SYSTEMS, IEEE AC-13,4 (1968)

(This research has been supported by Project THEMIS N00014-68-A-0152.)

ON SYSTEMS WITH UNKNOWN OR UNCERTAIN PARAMETERS

JOHN B. THOMAS

Department of Electrical Engineering
Princeton University, Princeton, N.J. 08540

ABSTRACT

This paper considers the analysis and synthesis of several classes of systems containing uncertain or unknown parameters which may be modeled as random variables. Attention is restricted primarily to linear, time-invariant systems and to simple mean-squared error criteria. The analysis problem is straightforward but computationally difficult. The synthesis problem is tractable only when the structure of the system containing the random parameters is well-specified.

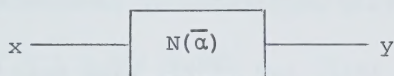
I. INTRODUCTION

A large portion of the literature of analysis and synthesis is devoted to those systems which are linear and time invariant. Such systems are often close approximations to reality and are relatively easy to analyse. Moreover, two quite different theories of synthesis for these systems have been developed extensively. The first, usually called network synthesis, is concerned with the approximation of arbitrary driving point impedances and complex transfer functions and relies heavily on complex variable and transform techniques and on topological methods. The second theory, frequently called linear optimization or linear optimal filtering, is concerned with the optimization of a linear system with random inputs, using as a criterion least mean-squared error or maximum output S/N ratio. The mathematical tools used are the second-moment theory of random processes and such optimization techniques as the calculus of variation. The interaction between these two areas of synthesis is quite small and is usually limited to the use of techniques of network synthesis to approximate unrealizable transfer functions which have been found by optimization methods.

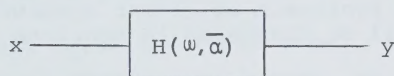
This paper is concerned with systems containing components or parameters whose values are unknown or uncertain. This situation may arise from a variety of causes: (1) components may be known only to within some tolerance range, (2) components may change in value or drift due to aging, temperature variation, overloading, etc., (3) random phenomena such as fading in a propagation channel may affect some part or parts of the system. Strictly speaking, not all of these uncertainties need be random in nature, although it may be convenient and enlightening to treat them as such.

The situation to be considered is shown in Fig. 1(a) where x is the input and y the output of a system $N(\bar{\alpha})$ which contains a set of q random parameters or random variables $\bar{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_q)$. In general x and y are functions of time although they may be either discrete or continuous in time and may be deterministic or may be realizations of random processes. In principle, the system N may be completely arbitrary, but it will usually be assumed that it is linear and time invariant with complex trans-

fer function $H(\omega, \bar{\alpha})$ and impulse response $h(t, \bar{\alpha})$, where ω is radian frequency and t is time. Such a system is shown in Fig. 1(b). Let the nominal, or ideal, or zero tolerance value (ZTV) of each random variable α_i be denoted by α_i' . In many cases α_i' will be the mean value $E\{\alpha_i\}$ but, in other cases, it may prove convenient to use other moments or characteristics of the density of α_i as the ZTV α_i' .



(a) a general system



(b) a linear time-invariant system

Fig. 1 - Systems with Random Parameters.

The ZTV of $H(\omega, \bar{\alpha})$ is $H(\omega, \bar{\alpha}')$ and, as the notation implies, will usually correspond to the value that $H(\omega)$ would assume if the set of parameters were fixed at their desired or nominal values. If one of the α_i is actually an element of a lumped linear network, then $H(\omega, \bar{\alpha})$ is a bilinear function of that α_i and may be written more explicitly as

$$H(\omega, \bar{\alpha}) = \frac{A(\omega) + \alpha_i B(\omega)}{C(\omega) + \alpha_i D(\omega)} \quad (1)$$

The function $H(\omega, \bar{\alpha})$ is a random variable for each value of ω and such operations as

$$u_H = E\{H(\omega, \bar{\alpha})\} = \int_{\bar{\alpha}} H(\omega, \bar{\alpha}) p(\bar{\alpha}; \bar{\alpha}') d\bar{\alpha} \quad (2)$$

are well-defined. Even when $E\{\alpha_i\} = \alpha_i'$ for all i , it will generally not be true that $u_H = H(\omega, \bar{\alpha}')$. However, it will frequently be desirable to compare $H(\omega, \bar{\alpha})$ with $H(\omega, \bar{\alpha}')$. To this end, one might consider one of the following measures of error:

- (i) the absolute difference $|H(\omega, \bar{\alpha}') - u_H|$.
- (ii) the variance of the random variable $H(\omega, \bar{\alpha})$, i.e.,

$$\sigma_H^2 = E\{[H(\omega, \bar{\alpha}) - u_H]^2\}.$$
- (iii) the averaged mean-squared error (mse) given by

$$\overline{\epsilon_a^2}(\bar{\alpha}') = E_{\alpha, x, y} \{(y - z)^2\}$$

where y is the actual output due to a random input x and z is a desired output. For example, z might be the output where $H(\omega, \bar{\alpha}) = H(\omega, \bar{\alpha}')$.

- (iiii) the mean-integral-squared error given by

$$E\left\{\int_a^b |k(t, \bar{\alpha}) - k(t, \bar{\alpha}')|^2 dt\right\}$$

where $k(t, \bar{\alpha})$ is the response of the system $H(\omega, \bar{\alpha})$ to some standard signal and $y(t, \bar{\alpha}')$ is the corresponding response of the ZTV system $H(\omega, \bar{\alpha}')$.

The actual measure used should be that one which is most appropriate to the given problem, but it will often be true that the choice will be heavily influenced by considerations of tractability.

Criteria (i), (ii), and (iii) have been used recently in statistical approaches to system sensitivity [1,2,3]. This subject will be discussed briefly in the next section. Criterion (iii) may be viewed as a generalization of the least-mean-squared error analysis and synthesis of linear systems. Various aspects of this subject will be discussed in Sections III and IV.

II. SYSTEM SENSITIVITY

The sensitivity $S_{\alpha_i}^H$ of a system $H(\omega, \bar{\alpha})$ to variations of a particular parameter α_i is defined as

$$S_{\alpha_i}^H = \frac{\partial}{\partial \ln \alpha_i} \ln H = \frac{\alpha_i}{H} \frac{\partial H}{\partial \alpha_i} \quad (3)$$

The usual approach for a nonrandom parameter α_i is to assume a small change $\Delta \alpha_i$ in the parameter and to consider the resulting change ΔH in $H(\omega, \bar{\alpha})$. To a first approximation we have

$$\frac{\Delta |H|}{|H|} \approx (\operatorname{Re} S_{\alpha_i}^H) \frac{\Delta \alpha_i}{\alpha_i} \quad (4)$$

and

$$\Delta \arg H \approx (\operatorname{Im} S_{\alpha_i}^H) \frac{\Delta \alpha_i}{\alpha_i} \quad (5)$$

where $\Delta |H|$ and $\Delta \arg H$ are the change in $|H|$ and in $\arg H$ corresponding to $\Delta \alpha_i$.

Suppose that α_i is a random variable which may be written as $\alpha_i = \alpha_i' + \delta_i$ where α_i' is the ZTV of α_i and δ_i is a random variable with zero mean and variance $\sigma_i^2 \ll \alpha_i'^2$. In this case the expected values of Eqs. (4) and (5) are approximately zero and the variances are given approximately by

$$E_{\alpha_i} \left\{ \left[\frac{\Delta |H|}{|H|} \right]^2 \right\} \approx (\operatorname{Re} S_{\alpha_i}^H)^2 \sigma_i^2 \quad (6)$$

and

$$E_{\alpha_i} \left\{ [\Delta \arg H]^2 \right\} \approx (\operatorname{Im} S_{\alpha_i}^H)^2 \sigma_i^2 \quad (7)$$

Thus, for a small random perturbation of the parameter α_i about its ZTV α_i' , the variance of the relative change in H is proportional to the variance of the perturbation. If all of the α_i are randomly perturbed in the same manner, the effects are

approximately additive [3], for example

$$E_{\alpha} \left\{ \left[\frac{\Delta |H|}{|H|} \right]^2 \right\} \approx \sum_i (\operatorname{Re} S_{\alpha_i}^H)^2 \sigma_i^2 \quad (8)$$

Sensitivity measures of the form of criterion (iiii) have been proposed [2] to reduce computational complexity. For simplicity, consider the single parameter case ($\bar{\alpha}=\alpha$). The error $e(t, \alpha) = k(t, \alpha) - k(t, \alpha')$ in the system response is expanded in some complete set of orthogonal functions $\{\varphi_i(\alpha)\}$ of the parameters α , i.e.,

$$e(t, \alpha) = \sum_i C_i(t) \varphi_i(\alpha) \quad (9)$$

where

$$E_{\alpha} \{\varphi_i(\alpha) \varphi_j(\alpha)\} = K \delta_{ij} \quad (10)$$

and the coefficients $C_i(t)$ are found from

$$C_i(t) = \frac{1}{K} E_{\alpha} \{e(t, \alpha) \varphi_i(\alpha)\} \quad (11)$$

Now the mean-integral-squared error becomes

$$E_{\alpha} \left\{ \int_a^b |e(t, \alpha)|^2 dt \right\} = K \int_a^b \sum_i C_i^2(t) dt \quad (12)$$

The right side of this last expression may be easier to evaluate than the left, depending on the form of the system response and on the set $\{\varphi_i(\alpha)\}$.

Some examples have been given [2] for simple systems described by second-order differential equations and a uniform distribution for α . The set $\{\varphi_i(\alpha)\}$ used in this case are the Walsh functions and it is claimed that the sum $\sum_i C_i^2(t)$ converges rapidly. It is clear that the choice of the $\{\varphi_i\}$ is critical. Also the rapidity of convergence of the sum $\sum_i C_i^2(t)$ would not appear easily to establish in the general case.

III. OPTIMAL LINEAR SYNTHESIS - FIXED CONFIGURATION

The least mean-squared error synthesis of linear systems of fixed configuration with respect to a finite number of parameters is well-known [4,5] and was one of the earliest examples of statistical optimization techniques. A linear system is assumed fixed except for the values of q parameters $\bar{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_q)$. Figure 1(b) will serve as a model of the situation with $H(\bar{\omega}, \bar{\alpha})$ a linear system, x a random process, and the $\bar{\alpha}$ a set of q undetermined parameters. The mean-squared error between the actual output y and the desired output z is given by

$$E\{(y-z)^2\} = \bar{\epsilon}^2(\bar{\alpha}) \quad (13)$$

and is a function of the q parameters $\bar{\alpha}$. Formally, there is no difficulty in calculating $\bar{\epsilon}^2(\bar{\alpha})$ when H is a lumped linear system and q is not too large ($q \leq 7$). In fact, the values of certain integrals which occur frequently in these calculations have been tabulated [4]. The optimum values $\bar{\alpha}_0 = (\alpha_{10}, \alpha_{20}, \dots, \alpha_{q0})$ to make

$\bar{\epsilon}^2(\bar{\alpha})$ a minimum can then be found by standard minimization techniques. The result is the minimum mean-squared error system $H(\omega, \bar{\alpha}_0)$ with least mean-squared error $\bar{\epsilon}^2(\bar{\alpha}_0)$. In general, increasing q will decrease $\bar{\epsilon}^2(\bar{\alpha}_0)$. Also, for fixed q , changing the basic configuration of H will change $\bar{\epsilon}^2(\bar{\alpha}_0)$.

For systems with non-ideal components, it is assumed that the q parameters $\bar{\alpha}$ are random variables with ZTV denoted by $\bar{\alpha}'$ and joint density $p(\bar{\alpha})$. It will frequently be possible to write this joint density as an explicit function of the $\bar{\alpha}'$. In this case the joint density will be denoted by $p(\bar{\alpha}; \bar{\alpha}')$. The mse $\bar{\epsilon}^2(\bar{\alpha})$ is now a random variable with expectation or mean value

$$E\{\bar{\epsilon}^2(\bar{\alpha})\} = \bar{\epsilon}_a^2(\bar{\alpha}') = \int_{\bar{\alpha}} \bar{\epsilon}^2(\bar{\alpha}) p(\bar{\alpha}; \bar{\alpha}') d\bar{\alpha} \quad (14)$$

If this "average" mse is accepted as the new measure of system performance, then Eq. (14) is minimized with respect to the q nominal or ZTV's $\bar{\alpha}' = (\alpha'_1, \alpha'_2, \dots, \alpha'_q)$. In principle, this problem is no more complicated than the minimization of $\bar{\epsilon}^2(\bar{\alpha})$ with respect to the $\bar{\alpha}$. In practice, the complexity of the minimization will depend on the form of the joint density $p(\bar{\alpha}; \bar{\alpha}')$. If it can be assumed that the α_i are statistically independent, then the joint density $p(\bar{\alpha}; \bar{\alpha}')$ can be factored and Eq. (14) becomes

$$\bar{\epsilon}_a^2(\bar{\alpha}') = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \bar{\epsilon}^2(\bar{\alpha}) \prod_{i=1}^q p_i(\alpha_i; \alpha'_i) d\alpha_1 \dots d\alpha_q \quad (15)$$

In general, this last expression will be easier to minimize than Eq. (14).

Example No. 1 - Consider the servomechanism shown in Fig. 2. A simple compensation network $G_1(\omega)$ of fixed configuration is to be designed. The input $\theta_i(t)$ is a series of step displacements of random amplitudes occurring at random time intervals. The random process defined by the time derivative of this input is a white noise process with constant power spectral density S_0 . The additive noise $\theta_n(t)$ is assumed to be an independent white noise with constant power spectral density N_0 . The motor

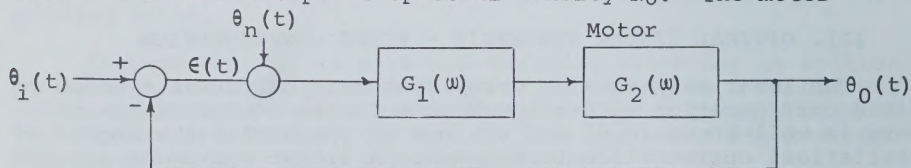


Fig. 2 - A Servomechanism with Compensating Network $G_1(\omega)$.

transfer function $G_2(\omega)$ is given by

$$G_2(\omega) = \frac{K_3}{j\omega(j\omega J + K_2)} \quad (16)$$

The compensation network has a fixed time constant T and an adjustable gain K_1 so that

$$G_1(\omega) = \frac{K_1}{1+j\omega T} \quad (17)$$

It is desired to find the value of K_1 which minimizes the mse

$\overline{\epsilon^2}(t)$. This problem is well-known [4] and the mse can be written as

$$\overline{\epsilon^2}(t) = \overline{\epsilon^2}(\alpha) = K_0 \frac{1+\alpha(1-A)+L\alpha^2}{\alpha(1-A\alpha)}, \quad 0 < \alpha < \frac{1}{A} \quad (18)$$

where

$$\alpha = (T+J/K_2) K_1 K_3 / K_2 \quad (19)$$

$$A = \frac{T(J/K_2)}{(T+J/K_2)^2} \quad (20)$$

$$L = \frac{N_0}{S_0(T+J/K_2)^2} \quad (21)$$

$$K_0 = (T+J/K_2) S_0 / 4 \quad (22)$$

Note that α is a linear function of K_1 and that all of the other parameters are known. If α is outside the interval $(0, 1/A)$, the system is unstable and the mse is infinite. Equation (18) is easily minimized with respect to α . The result is

$$\overline{\epsilon^2}(\alpha_0) = K_0 [A+2(A+L)^{1/2} + 1] \quad (23)$$

where the optimal value of α is α_0 given by

$$\alpha_0 = [A+(A+L)^{1/2}]^{-1} \quad (24)$$

Let us now assume that α is a random variable uniformly distributed about α' so that

$$p(\alpha, \alpha') = \begin{cases} 1/2\beta & , \quad \alpha' - \beta \leq \alpha \leq \alpha' + \beta \\ 0 & , \quad \text{elsewhere} \end{cases} \quad (25)$$

where β is a constant. Here the nominal or ZTV α' is the mean of α . The average mse is easily calculated to be

$$\overline{\epsilon_a^2}(\alpha') = \frac{K_0}{2\beta} \left[\left(\frac{L}{A} + \frac{1}{A} \right) \ln \frac{1-A(\alpha'-\beta)}{1-A(\alpha'+\beta)} + \ln \frac{\alpha'+\beta}{\alpha'-\beta} - \frac{2L\beta}{A} \right] \quad (26)$$

This last expression is a minimum when α' has its optimal value α_0' given by

$$\alpha_0' = \frac{A - [(A+L) + \beta^2(A^2 - A - L)^2]^{1/2}}{A^2 - A - L} \quad (27)$$

where α_0' is constrained to satisfy the inequality

$$\beta \leq \alpha_0' \leq \frac{1}{A} - \beta \quad (28)$$

when $\beta=0$, then $\alpha_0' = \alpha_0$ and $\overline{\epsilon_a^2}(\alpha_0') = \overline{\epsilon^2}(\alpha)$ as expected.

Let us assume that the two time constants T and J/K_2 are equal so that the parameter A has the value 0.25. Let the ratio of S_0 to N_0 be 4.0 and let the time constants lie in the range 0.05 to 1.0 seconds. Now the parameter L is in the range 0.0625 to 25.0. We would expect that the minimum average mean-squared error $\overline{\epsilon_a^2}(\alpha_0')$ would be an increasing function of β , the half-width of the density function. Figure 3 shows the percentage increase in minimum average mean-squared error as a function of β and for various values of the parameter L . Further details

may be found elsewhere [6].

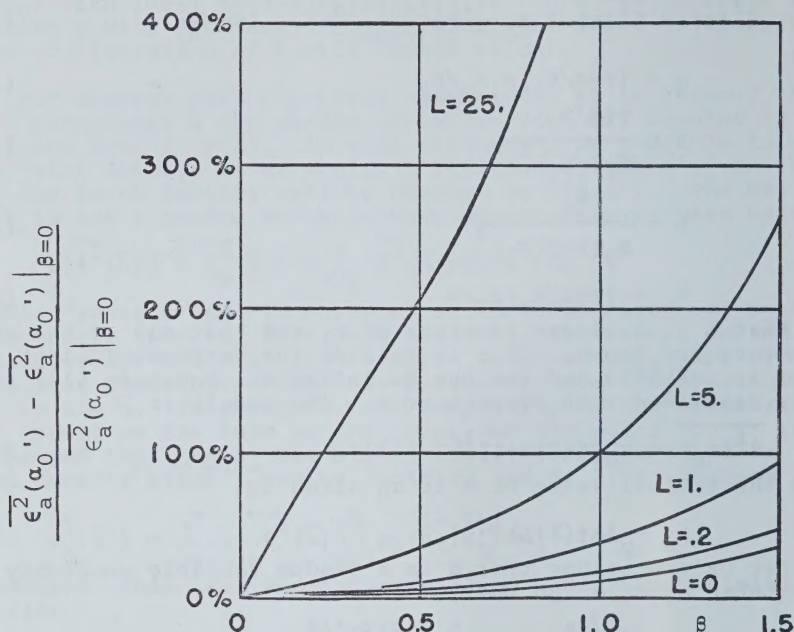


Fig. 3 - Percentage increase in minimum average mse as a function of half-width of density function

We may wish to know the conditions for which $\bar{\alpha}_0 = \bar{\alpha}_0'$. In this case the optimal choice of the nominal value of each random variable α_i is numerically equal to the optimal choice of the non-random parameter α_i . It can be shown [6] that sufficient, but not necessary, conditions are that (1) each density function $p_i(\alpha; \alpha')$ is symmetric in α about α' , and that (2) the mse $\epsilon^2(\bar{\alpha})$ is symmetric in $\bar{\alpha}$ about its minimum $\epsilon^2(\bar{\alpha}_0)$.

IV. UNCONSTRAINED OPTIMAL LINEAR SYNTHESIS

In the notation of Fig. 1(a) and of Section III, if y is to be a least mean-squared estimate of z , it is well-known that the system N is determined from the relationship $E\{\delta y(y-z)\} = 0$, where δy is an arbitrary variation in y , or, equivalently, from the orthogonality principle $E\{y(y-z)\} = 0$. For our purposes, either relationship is satisfactory and both imply that y is determined by the conditional expectation

$$y = E\{z|x, \bar{\alpha}\} \quad (29)$$

If the system contains no random variables $\bar{\alpha}$, then Eq. (29) reduces to the usual

$$y = E\{z|x\} \quad (30)$$

In general, it is difficult to find the least mean-squared system specified by Eq. (30) and most approaches to the problem

involve the addition of some constraints on the form of the system, e.g., that it be linear. In such cases, Eq. (30) is not satisfied, of course, and a system with a larger mean-squared error is actually determined. For our discussion, however, Eq. (29) poses a further difficulty. When random parameters are present in the system, it will not be possible to find the conditional expectation of Eq. (29) unless the statistics of the $\bar{\alpha}$ are known and unless the functional dependence of the conditional expectation on the $\bar{\alpha}$ is known. This amounts to requiring specific restrictions on the form of the system with respect to the $\bar{\alpha}$ and destroys the underlying assumption that the system is unconstrained.

We clarify the discussion of the last paragraph by restricting attention to linear systems of the type shown in Fig. 1(b) described by the complex transfer function $H(\omega, \bar{\alpha})$ or the impulse response $h(t, \bar{\alpha})$. Thus the input x and the output y are related through the convolution $y=h*x$ and the specifying equation $E\{\delta y(y-z)\}=0$ becomes

$$E\{\delta h*x(h*x-z)\} = 0 \quad (31)$$

If the linear system h contains no random parameters, this expression becomes the well-known Wiener-Hopf equation

$$E\{xh*x-zx\} = 0 \quad (32)$$

The linear filter problem defined by Eq. (31) is intractable for the same reasons as were discussed in conjunction with Eq. (29). Most attempts to deal with the problem [7,8,9] have involved making the assumption, either explicitly or implicitly, that the linear filter h can be written as the series combination of two filters h_1 and h_2 . The filter h_1 is assumed to contain the random parameters $\bar{\alpha}$ and to be fixed in form and in functional dependence on the $\bar{\alpha}$ so that statistical operations such as $E\{h_1\}$ are well-defined. In this case h is the convolution h_1*h_2 and the variation δh becomes $h_1*\delta h_2$. Now Eq. (31) becomes tractable since it can be written as

$$\delta h_2 * E\{h_1 * x(h_1 * h_2 * x - z)\} = 0 \quad (33)$$

or, if the $\bar{\alpha}$ are statistically independent of x and z and δh_2 is an arbitrary variation,

$$E\{h_1 * h_1\} * h_2 * R_{xx} - E\{h_1\} * R_{xz} = 0 \quad (34)$$

where R_{xx} is the autocorrelation function of x and R_{xz} is the cross-correlation between x and z .

If we ignore physical realizability and restrict attention to continuous parameter random processes x and z which are at least wide-sense stationary, then this last expression is easily solved for $H_2(\omega)$, the complex transfer function corresponding to h_2 , to yield

$$H_2(\omega) = \frac{E\{H_1(\omega)\} \varphi_{xz}(\omega)}{E\{H_1(\omega)H_1^*(\omega)\} \varphi_{xx}(\omega)} \quad (35)$$

where $\varphi_{xy}(\omega)$ and $\varphi_{xx}(\omega)$ are power spectral densities and $H_1(\omega)$ is the complex transfer function of filter h_1 . If the term $E\{H_1(\omega)H_1^*(\omega)\}$ can be factored in the form

$$E\{H_1(\omega)H_1^*(\omega)\} = G_1(\omega)G_1^*(\omega) \quad (36)$$

then, in the usual way, the realizable expression for $H_2(\omega)$ is given by

$$H_2^+(\omega) = \frac{1}{2\pi G_1(\omega)\varphi_{xx}^+(\omega)} \int_0^\infty e^{-j\omega t} \int_{-\infty}^\infty \frac{E\{H_1(\nu)\varphi_{xz}(\nu)\}}{G_1^*(\omega)\varphi_{xx}^-(\omega)} e^{j\nu t} d\nu dt \quad (37)$$

Here the input power spectral density $\varphi_{xx}(\omega)$ has been factored as $\varphi_{xx}(\omega) = \varphi_{xx}^+(\omega)\varphi_{xx}^-(\omega)$ where $\varphi_{xx}^+(\omega)$ has no lower-half plane singularities, $\varphi_{xx}^-(\omega)$ has no upper-half plane singularities and $\lambda = \omega + j\sigma$.

Example No. 2 - For the usual Wiener filter problem, $H_1(\omega) = G_1(\omega) = 1$ and the solution is well known.

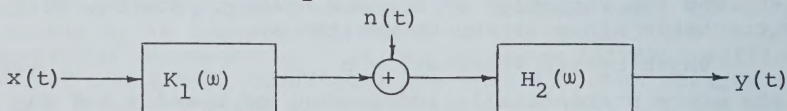
Example No. 3 - Consider the system shown in Fig. 4 where a noise $n(t)$ is added after the filter $K_1(\omega)$. In this case, $K_1(\omega)$ may represent a channel containing random parameters. Let us assume that the desired output $z(t)$ is a translated estimate of the input so that $z(t) = x(t+T)$. It is clear that Figs. 4(a) and 4(b) are equivalent in so far as input-output relationships are concerned, where $\varphi_{nn}(\omega)$ is the power spectral density of the additive noise $n(t)$. It follows from Eq. (35) that the unrealizable $H_2(\omega)$ is

$$H_2(\omega) = \frac{\frac{E\{H_1(\omega)\}\varphi_{xx}(\omega)e^{j\omega T}}{\alpha}}{\frac{E\{H_1(\omega)H_1^*(\omega)\}\varphi_{xx}(\omega) + \varphi_{nn}(\omega)}{\alpha}} \quad (38)$$

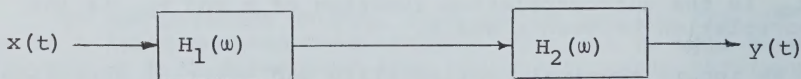
where $K_1(\omega)$ and $H_1(\omega)$ are related through

$$|H_1(\omega)|^2 = |K_1(\omega)|^2 + \varphi_{nn}(\omega)/\varphi_{xx}(\omega) \quad (39)$$

The realizable filter $H_2(\omega)$ can be written from Eq. (37).



a) Basic system - $K_1(\omega)$ is a random channel



b) Equivalent system when $|H_1(\omega)|^2 = |K_1(\omega)|^2 + \varphi_{nn}(\omega)/\varphi_{xx}(\omega)$

Fig. 4 - Optimal Filtering Involving a Random Channel

Suppose the mean-integral-squared-error (mise) criterion (iiii) of Section I is applied to an unconstrained linear system. The optimal mise system is then found as a solution to the equation

$$E\{A[\delta y(y-z)]\} = 0 \quad (40)$$

As before, y is the output, z is the desired output, and $A[\cdot]$ indicates the time average operation

$$A[\cdot] = \int_a^b [\cdot] dt \quad (41)$$

For systems which are constrained to be linear, Eq. (40) becomes

$$E\{A[\delta h * x(h * x - z)]\} = 0 \quad (42)$$

Example No. 4 - Consider again the system of Fig. 4(a). Let the input $x(t)$ be a deterministic test function with Fourier transform $X(\omega)$; let $n(t)$ again be a zero-mean, wide-sense stationary random process with autocorrelation function $R_{nn}(\tau)$ and power spectral density $\varphi_{nn}(\omega)$; let the filter $H_2(\omega) = \mathcal{F}\{h_2(t)\}$ be found so that the desired output z is a mise estimate of the random linear channel $k_1(t)$. Thus, we place $z(t) = k_1(t)$ with Fourier transforms $Z(\omega) = K_1(\omega)$. Now Eq. (42) can be solved since

$$y = h * x = h_2 * (k_1 * x + n) \quad (43)$$

and

$$\delta y = \delta h * x = \delta h_2 * (k_1 * x + n) \quad (44)$$

On making use of the fact that $E\{n\} = 0$, these last two equations allow Eq. (42) to be written as

$$\delta h_2 * \left[E\{A[(h_2 * k_1 * x)(k_1 * x)]\} + E\{h_2 * R_n\} - E\{A[k_1(k_1 * x)]\} \right] = 0$$

Since δh_2 is arbitrary, this last expression implies that the term with which δh_2 is convolved must be zero.

Let us now assume that the response $k_1(t)$ of the random channel is essentially contained in the interval (a, b) ; thus the limits a and b on the time averaging integral can be replaced with $-\infty$ and ∞ , respectively. If we now ignore physical realizability and transform freely, the term $[\cdot]$ in $\delta h_2 * [\cdot]$ becomes

$$H_2(\omega) E\{|K_1(\omega)|^2\} |X(\omega)|^2 + H_2(\omega) \varphi_n(\omega) - E\{|K_1(\omega)|^2\} X^*(\omega) = 0$$

On solving for $H_2(\omega)$, we have the well-known result [8] that

$$H_2(\omega) = \frac{E\{|K_1(\omega)|^2\} X^*(\omega)}{E\{|K_1(\omega)|^2\} |X(\omega)|^2 + \varphi_n(\omega)} \quad (45)$$

For vanishingly small noise $[\varphi_n(\omega) \rightarrow 0]$, this last expression reduces to

$$H_2(\omega) = \frac{1}{X(\omega)} \quad (46)$$

as expected. In the small signal case, we can assume that $E\{|K_1(\omega)|^2\} |X(\omega)|^2 \ll \varphi_n(\omega)$ and Eq. (45) reduces to

$$H_2(\omega) \approx \frac{X^*(\omega)}{\varphi_r(\omega)} \quad (47)$$

where $\varphi_r(\omega) = \varphi_n(\omega) / E\{|K_1(\omega)|^2\}$ is the input noise spectral density referred back through the average channel characteristics. If $\varphi_r(\omega)$ is constant, then Eq. (47) reduces to the usual matched filter obtained by maximizing the S/N.

V. REMARK

Certain classes of problems involving random parameters have not been discussed due to lack of space. One such class arises in the realization of digital filters where errors due to quantization, coefficient inaccuracy, and round-off accumulation can be treated as random [10,11].

ACKNOWLEDGMENT

The author wishes to acknowledge the support of the Air Office of Scientific Research under AFOSR Grant 1333-67, of the National Science Foundation under Grant GK-1439, and of the U.S. Army Research Office under Contract DAHCO4-69-C-0012.

REFERENCES

1. R.F. Whitback, "The Application of Linear Optimal Control to Systems Containing Uncertain Parameters," Joint Automatic Control Conference, Seattle, Washington, June 1966.
2. L. Shaw and P. Chalon, "An Expansion for Evaluation of Sensitivity with Respect to a Random Variable Parameter," Second IFAC Symposium Preprints, System Sensitivity and Adaptivity, Dubrovnik, August 26-31, 1968; pp. C59-C69.
3. B. Liu and M.E. Van Valkenburg, "A Probabilistic Approach to the Parameter Sensitivity Problem," Proceedings of the Fifth Annual Allerton Conference on Circuit and System Theory, October 4-6, 1967; pp. 701-710.
4. H.M. James, N.B. Nichols and R.S. Phillips, Theory of Servomechanisms, MIT Rad. Lab. Series, vol. 25, McGraw-Hill Book Company, New York, N.Y., 1947.
5. G.C. Newton, Jr., L.A. Gould and J.F. Kaiser, Analytic Design of Linear Feedback Controls, John Wiley & Sons, New York, N.Y., 1957.
6. J.K. Wolf, J.B. Thomas and T.R. Williams, "The Synthesis of Optimum Systems from Nonideal Components," AIEE Transactions (Applications and Industry), No. 55, July 1961; pp. 149-152.
7. R. Price, "Optimum Detection of Random Signals in Noise, with Application to Scatter-Multipath Communication," I, vol. IT-2, No. 4, December 1956; pp. 125-135.

8. G.J. Turin, "On the Estimation in the Presence of Noise of the Impulse Response of a Random, Linear Filter," IRE Transactions on Information Theory, vol. IT-3, No. 1, March 1957; pp. 5-10.
9. F.J. Beutler, "Prediction and Filtering for Random Parameter Systems," IRE Transactions on Information Theory, vol. IT-4, No. 4, December 1958; pp. 166-171.
10. J.B. Knowles and E.M. Olcayto, "Coefficient Accuracy in Digital Filter Response," IEEE Transactions on Circuit Theory, vol. CT-15, No. 1, March 1968; pp. 31-41.
11. T. Kaneko and B. Liu, "Round-Off Error of Floating-Point Digital Filters," Proceedings of the Sixth Annual Allerton Conference on Circuit and System Theory, October 2-4, 1968.

ADAPTATION IN A QUANTIZED PARAMETER SPACE

ALLEN GERSHO

Bell Telephone Laboratories, Incorporated

Murray Hill, New Jersey

ABSTRACT

An optimum-seeking algorithm is presented for an adaptive system whose adjustable parameters are constrained to assume only quantized values. Each iteration consists of moving from the current point in the quantized parameter space to one of the adjacent points determined by the polarities of the gradient components. Convergence is proved for smooth convex cost functions.

INTRODUCTION

A large class of adaptation problems fall into the model, illustrated in Fig. 1, consisting of a system containing a set of adjustable parameters $c_1, c_2, \dots, c_N$ and an "adaptor" or parameter adjusting mechanism. For a given parameter setting the system operates with a certain performance quality or cost determined by a cost function, $Q(c_1, c_2, \dots, c_N)$. The adaptor processes certain observable signals from the system obtaining correction signals to increment the parameter values with the object of minimizing or suitably reducing the cost. In the "continuous-time" case the parameters are continually adjusted (as in References 1 and 2), and the adaptation strategy is usually described by a first-order differential equation in the parameter vector $\underline{c} = (c_1, c_2, \dots, c_N)$. In the "discrete-time" case, of interest in this paper, the parameters are incremented at discrete time instants and the adaptation strategy is characterized by an iterative algorithm or, in other words, by a first-order difference equation in the parameter vector $\underline{c}$. (See, for example, References 3, 4 or 5.) Often an acceptable performance cost ϵ is specified so that convergence of the algorithm to the optimal (cost-minimizing) parameter setting is not necessary; it may suffice to reach an "admissible" region of the parameter space where the cost is below a threshold, ϵ , of tolerable cost.

The nature of the cost function and available observable signals are of course highly relevant to both the question of existence of an adaptation strategy for converging to the admissible region and the problem of determining a suitable strategy. In the most widely studied case the cost function is smooth and convex and the observable signals determine the gradient of the cost at the current point in the parameter space. The usual gradient method⁶ and the numerous variations provide useful convergent algorithms for adaptation in this situation.

In one important class⁷ of adaptive systems which includes most pattern-classifying machines, the adjustable parameters are weights used to form a new signal as a weighted sum of other signals. In this case a convenient and often used cost function is the mean square error between the actual weighted sum achieved and a desired signal. The cost function is then quadratic and convex and when the desired signal is known the adaptor can readily determine (or estimate) the gradient vector from observations of the signals being weighted and the weighted sum signal. In other more complex systems, however, the estimation of the gradient could be far more difficult.

An important consideration in the choice of optimum-seeking algorithms for a real-time adaptive system is the resulting implementation complexity. Relatively simple implementation using integrated logic circuitry becomes possible when the permissible parameter values are quantized and only digital information about the system operation is used as input data for the adaptor. Digital command signals then suffice to direct parameter adjustments and less information about the system operation is needed to generate correction signals. With this motivation an algorithm is presented here based on quantization of the gradient vector so that only the signs of the gradient components are needed as input data. For a special case where the cost function is piecewise linear an effective quantized-parameter algorithm has already been found,³ however, few results of a general nature are known. Most algorithms⁸⁻¹¹ considered are heuristic in nature, apply to single-parameter systems or involve adjustment of one parameter at a time, and usually require numerous evaluations of the cost function. The quantized gradient algorithm examined here is a natural digitalized version of the usual gradient methods and has the often desirable feature that the cost function itself need never be evaluated.

QUANTIZED PARAMETER SPACE

A parameter setting may be described by the vector $\underline{c}$ in the n -dimensional Euclidean parameter space $\mathbb{R}^N$. A quantization of the parameter space consists of replacing the space with a countable set $\mathcal{P}$ of points in $\mathbb{R}^N$ having the property that for some finite number $d > 0$, each point in $\mathbb{R}^N$ is less than a distance d from some point in $\mathcal{P}$. We shall consider only the simplest form of quantization - the uniform case where the points in $\mathcal{P}$ are the intersection points of an n -dimensional rectangular grid parallel to the coordinate axes with spacing δ between two adjacent parallel lines. An iterative algorithm for adaptation is then a strategy for moving from a current point in $\mathcal{P}$ to a new point in $\mathcal{P}$ starting from a specified initial point $\underline{c}_0$ in $\mathcal{P}$. We shall say the algorithm is convergent if in some finite number of steps $\bar{M}$ (the convergence time), the parameter vector reaches a point in the admissible region

$$Q \triangleq \{\underline{c}: Q(\underline{c}) < \varepsilon\} \quad (1)$$

and remains henceforth in this region. For convergence it is of course necessary that the intersection of the sets Q and $\mathcal{P}$ contain at least one point. This condition is satisfied for a particular cost function and threshold cost level $\varepsilon > 0$ when the quantization is sufficiently fine, i.e., when δ is sufficiently small.

In some applications of adaptation where the iterative process is stopped when the cost drops below ε , it is not necessary to require that the parameter vector remain in the admissible region with continued application of the iterative algorithm. However, in other situations this requirement is essential as, for example, when the algorithm is applied to tracking adaptation where system characteristics are nonstationary.

GRADIENT-POLARITY ALGORITHM

Motivated by the usual gradient method, we consider the digitalized algorithm where the parameter corrections are based on the crudest

and simplest quantization of the gradient vector. Specifically the algorithm is:

$$\underline{c}_{n+1} = \underline{c}_n - \delta \operatorname{sgn} \underline{g}(\underline{c}_n) \quad (2)$$

where $\underline{c}_0$ is in $\mathcal{P}$ and

$$\underline{g}(\underline{c}) = \underline{\nabla} Q(\underline{c}) \quad (3)$$

and

$$\operatorname{sgn} \underline{g} = (\operatorname{sgn} g_1, \operatorname{sgn} g_2, \dots, \operatorname{sgn} g_N). \quad (4)$$

Thus for each parameter setting we need only know whether the cost decreases or increases with an incremental change in the k^{th} parameter for each k in order to determine the new parameter setting. Each component of the parameter vector is incremented by the amount $\pm\delta$ according to whether the corresponding gradient component is negative or positive.

We shall assume that the cost function has continuous second partial derivatives, the Hessian $H(\underline{c})$ is everywhere positive definite, and for all points in $\mathcal{R}^N$ the minimum eigenvalue of $H(\underline{c})$ is never less than $\lambda > 0$ and the maximum eigenvalue never exceeds $\Lambda < \infty$. Furthermore we assume that there exists a point where the gradient is zero and the cost has a minimum. Then without loss of generality we may assume that the minimum occurs at the origin and at this point the cost is zero, i.e.,

$$\underline{\nabla} Q(\underline{0}) = \underline{0} \quad \text{and} \quad Q(\underline{0}) = 0. \quad (5)$$

Note that we have not assumed that the origin is contained in $\mathcal{P}$.

The main result of this paper is a proof that the algorithm given by (1) is convergent for a suitably fine quantization of the parameter space. For convenience we restate the result below in theorem form.

Theorem

Let $Q(\underline{c})$ be a cost function in $\mathcal{C}^2$ with $\underline{c} \in \mathcal{R}^N$, $Q(\underline{0}) = 0$, $\underline{\nabla} Q(\underline{0}) = \underline{0}$, $H(\underline{c})$ the Hessian of Q positive definite with spectrum bounded by

$$0 < \lambda \leq \operatorname{Eig}\{H(\underline{c})\} \leq \Lambda \quad (6)$$

for all $\underline{c}$. Let $\underline{g}(\underline{c}) = \underline{\nabla} Q(\underline{c})$ and $\widehat{g}(\underline{c}) = \operatorname{sgn} \underline{g}(\underline{c})$. For any $\underline{c}_0$ let $\underline{c}_{k+1} = \underline{c}_k - \delta \widehat{g}(\underline{c}_k)$, $k = 0, 1, 2, \dots$. Then given a cost threshold $\varepsilon > 0$, there exists a quantization size $\delta > 0$ such that for some $M > 0$, $Q(\underline{c}_k) < \varepsilon$ for all $k \geq M$.

PROOF OF CONVERGENCE

The convergence of the algorithm (2) as described by the above theorem may be seen by first developing some simple lemmas. First, if $|\underline{g}|$ and $\|\underline{g}\|$ denote respectively the usual ℓ_1 and ℓ_2 norms (summed magnitudes and root summed squares of the components) of the n-tuple vector $\underline{g}$, we have the following easily-verified inequality:

Lemma 1

$$\|\underline{g}\| \leq |\underline{g}| \leq \sqrt{N} \|\underline{g}\|.$$

Now by Taylor's theorem applied to $Q(\underline{c})$ about the origin using (5), we obtain

$$Q(\underline{c}) = \frac{1}{2} \underline{c}' H_{\theta} \underline{c} \quad (7)$$

where $H_{\theta} = H(\theta \underline{c})$ for some θ with $0 \leq \theta \leq 1$. This expression for the cost is bounded by

$$Q(\underline{c}) \leq \frac{1}{2} \Lambda \|\underline{c}\|^2 \quad (8)$$

using (6). Furthermore application of Taylor's theorem to $\underline{c}' \underline{g}(t \underline{c})$ with respect to t yields for $t = 1$:

$$\underline{c}' \underline{g}(\underline{c}) = \underline{c}' H(\alpha \underline{c}) \underline{c} \quad (9)$$

so that, using (6) and Schwarz's inequality, we have

$$\|\underline{c}\| \|\underline{g}\| \geq \underline{c}' H_{\alpha} \underline{c} \geq \lambda \|\underline{c}\|^2 \quad (10)$$

and

$$\|\underline{c}\| \leq \frac{1}{\lambda} \|\underline{g}\|. \quad (11)$$

Applying (11) to (8) yields the result

Lemma 2

$$Q(\underline{c}) \leq \frac{\Lambda}{2\lambda^2} \|\underline{g}\|^2.$$

Applying Taylor's theorem about the point $\underline{c}_{n+1}$, using the algorithm (2) we find that

$$Q_{n+1} = Q_n - \delta \underline{g}'_n \hat{\underline{g}}_n + \frac{1}{2} \delta^2 \hat{\underline{g}}_n' H_{\theta} \hat{\underline{g}}_n \quad (12)$$

where $Q_n \triangleq Q(\underline{c}_n)$ and $\underline{g}_n = \underline{g}(\underline{c}_n)$. Noting that $\hat{\underline{g}}_n' \hat{\underline{g}}_n = |\underline{g}_n|$ yields

$$Q_n - Q_{n+1} = \delta \left[|g_n| - \frac{1}{2} \delta \hat{g}_n' H_{\theta} \hat{g}_n \right], \quad (13)$$

but

$$\hat{g}_n' H_{\theta} \hat{g}_n \leq \Lambda \|\hat{g}_n\|^2 = N\Lambda \quad (14)$$

with the result

Lemma 3

$$Q_n - Q_{n+1} \geq \delta(|g_n| - b)$$

where

$$b \triangleq \frac{1}{2} \delta N\Lambda. \quad (15)$$

From Lemma 3 we have

$$Q_{n+1} \leq Q_n + \delta b$$

but from Lemmas 2 and 1,

$$Q_n \leq \frac{\Lambda}{2\lambda} \|g_n\|^2 \leq \frac{\Lambda}{2\lambda^2} |g_n|^2 < \frac{\Lambda}{2\lambda^2} b^2 \quad (16)$$

if $|g_n| < b$, so

$$Q_{n+1} \leq \frac{\Lambda}{2\lambda^2} b^2 + \delta b \quad (17)$$

but

$$\delta b < 8 \left(\frac{\Lambda}{2\lambda^2} \right) b^2 \quad (18)$$

so that we have

Lemma 4

$$\text{If } |g_n| < b \text{ then } Q_{n+1} < \frac{9\Lambda}{2\lambda^2} b^2$$

Finally we note that $|g| < 3b$ implies, using Lemma 1, that $\|g\| < 3b$ and therefore Lemma 2 yields

Lemma 5

$$\text{If } |g| < 3b \text{ then } Q < \frac{9\Lambda}{2\lambda^2} b^2.$$

To prove convergence of the algorithm (2), choose δ to satisfy the relation

$$\varepsilon = \frac{9\Lambda}{2\lambda^2} b^2 \quad (19)$$

Now suppose $|g_n| \geq 3b$ (for otherwise Lemma 5 implies $Q_n < \varepsilon$ and we are initially in the admissible region). Then from Lemma 3 we see that $|g_n|$ must drop below $3b$ for otherwise Q_n will drop at an arithmetic rate to $-\infty$ which is impossible. Thus for some integer $M < \infty$, we must have $|g_M| < 3b$, and so by Lemma 5, $Q_M < \varepsilon$ and c_M is in the admissible region $\bar{Q}$. To show that c remains henceforth in $\bar{Q}$ note that for any $n \geq M$, if $|g_n| \geq b$ then $Q_{n+1} < Q_n$ by Lemma 3, and if $|g_n| < b$, then Lemma 4 requires that $Q_{n+1} < \varepsilon$. Then, by induction, $Q_n < \varepsilon$ for all $n \geq M$ which completes the proof.

Remark

A quantization size δ sufficiently small to assure convergence is found from (19) and (15) to be

$$\delta = \frac{2\lambda}{3N} \left(\frac{2\varepsilon}{\Lambda^3} \right)^{1/2}. \quad (20)$$

Furthermore a conservative upper bound on the convergence time $\bar{M}$ may be obtained from Lemma 3 by noting that the cost drops in each iteration by at least the amount $2b\delta$ so that

$$\bar{M} \leq \frac{Q_0}{2b\delta} = \frac{Q_0}{\delta^2 \Lambda N}. \quad (21)$$

Using (20) we obtain the useful bound

$$\bar{M} \leq \frac{9}{8} N \left(\frac{\Lambda}{\lambda} \right)^2 \frac{Q_0}{\varepsilon}. \quad (22)$$

This bound on convergence time may be very crude for many applications since the cost drops much more quickly in regions where the gradient is large than in the vicinity of the admissible region.

CONCLUDING REMARKS

The convergence of the quantized gradient algorithm proves that the sign of the gradient components at successive parameter settings gives sufficient information to the adaptor to achieve convergence to the admissible region. The price paid for the simplicity of the algorithm and resulting implementation is of course a rather slow rate of convergence. However the algorithm used here does not necessarily offer the best strategy for utilizing the gradient polarity information. For example it should be possible in each iteration to effectively make larger steps in the quantized parameter space by utilizing prior

as well as current gradient polarity information. Another generalization that can be expected to improve convergence speed is to suitably use more than a two-level quantization of the gradient vector and correspondingly to send more than a two-level correction signal to each parameter. The class of systems and cost functions for which the quantized gradient algorithm is useful could be substantially increased by utilizing methods of nonlinear programming to incorporate constraints on the usable region of parameter space. Finally, it seems that the results obtained in this paper can be applied to the problem of adaptation in a stochastic environment. In this case the gradient polarities may be estimated by a suitable hypothesis testing technique and conditions may be obtained for convergence in a stochastic sense.

REFERENCES

1. M. M. Sondhi, An Adaptive Echo Canceller, Bell Sys. Tech. J., Vol. 46, pp. 497-511, March 1967.
2. K. S. Narendra and L. E. McBride, Multiparameter Self-Optimizing Systems Using Correlation Techniques, IEEE Trans. Aut. Cont., Vol. AC-9, pp. 31-38, January 1964.
3. R. W. Lucky, Techniques for Adaptive Equalization of Digital Communication Systems, B.S.T.J., 45, pp. 255-286, February 1966.
4. B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, Adaptive Antenna Systems, Proc. IEEE, Vol. 55, No. 12, pp. 2143-2159, December 1967.
5. A. Gersho, Adaptive Equalization of Highly Dispersive Channels for Data Transmission, Bell Sys. Tech. J., to appear December 1968.
6. A. A. Goldstein, Constructive Real Analysis, Harper and Row, New York, 1967.
7. Y. A. Tsypkin, Adaptation, Training, and Self-Organization in Automatic Systems, Automation and Remote Control, Vol. 27, No. 1, pp. 23-61, January 1966.
8. Takashi Isobe, Self-Optimization of a Control System by Means of a Logic Circuit, IRE Trans. on Aut. Control, September 1961, pp. 260-266.
9. E. D. Avedyan, Modification of the Algorithm of Robbins and Munro, Automation and Remote Control, No. 4, pp. 165-167, April 1967.
10. I. I. Perlman, Method of Self-Adjustment of Step Search Systems, Automation and Remote Control, No. 4, pp. 80-93, April 1967.
11. V. W. Eveleigh, Adaptive Control and Optimization Techniques, McGraw-Hill, 1967, p. 193.

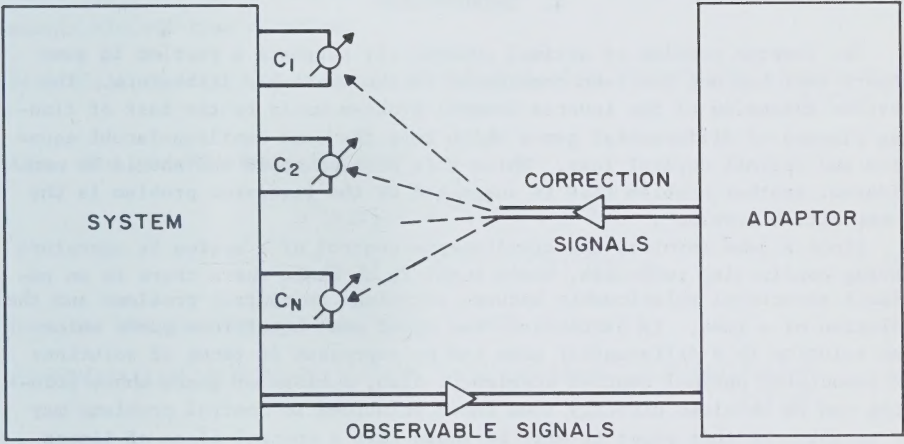


FIGURE I
MODEL OF A CLASS OF ADAPTIVE SYSTEMS

"THE COMPOSITION PROBLEM FOR DIFFERENTIAL GAMES"

MICHAEL D. CILETTI

Frank J. Seiler Research Laboratory (OAR)

US Air Force Academy, Colorado 80840

ABSTRACT

The task of finding a relationship between optimal control problems and differential games is presented as the "composition problem for differential games". Two classes of differential games are shown to have the property that their solution (the value function) can be constructed from solutions to associated (one-player) optimal control problems.

1. INTRODUCTION

The inverse problem of optimal control [1] suggests a problem in game theory that has not yet been considered in the available literature. The obvious extension of the inverse control problem would be the task of finding classes of differential games which have the same Hamilton-Jacobi equation and optimal control laws. While this problem could and should be considered, another problem that is suggested by the inversion problem is the "composition problem".

Since a game involves the simultaneous control of a system by operators having conflicting interests, there ought to be cases where there is an explicit structural relationship between solutions to control problems and the solution of a game. In particular, one might seek conditions under which the solution to a differential game can be expressed in terms of solutions to associated optimal control problems. Also, a class of games whose solution can be obtained directly from known solutions to control problems may be sought. In this paper it will be shown that a special class of linear pursuit-evasion games and a class of games for norm-invariant systems do have a composition property.

2. LINEAR PURSUIT-EVASION GAMES

The generalization of the state regulator problem of optimal control theory to a game-theoretic context is important in that it can be used [2] to describe a conflict between two systems in terms of their relative coordinates. The situation is described by a linear dynamical system:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + C(t)v(t), \quad x(t_0) = x_0, \quad (1)$$

where x is an n -vector describing the state of the system, A , B and C are $n \times n$, $n \times q$, and $n \times m$ matrices of continuous functions of t , respectively. The quantity u is a q -vector control for P , the pursuer, defined on $R^n \times R$, continuous in t and Lipschitzian in x , with $u(x, t) \in R^q$, and K_u denotes the class of all such $u(\cdot)$. Likewise, v is an m -vector control for E , the evader, defined on $R^n \times R$, continuous in t and Lipschitzian in x , with $v(x, t) \in R^m$, and K_v denotes the class of all such $v(\cdot)$.

The playing space is given by $G = \{(x, t): (x, t) \in R^n \times R, t \geq t_0\}$ and the

target set is $S = \{(x, t) : (x, t) \in R^n \times R, t = T \geq t_0\}$. Payoff to the evader is determined by the functional:

$$J(x_0, t_0, u, v) = 1/2 \langle x(T), Fx(T) \rangle + 1/2 \int_{t_0}^T [\langle x(\alpha), Q(\alpha)x(\alpha) \rangle + \langle u(x(\alpha), \alpha), R(\alpha)u(x(\alpha), \alpha) \rangle - \langle v(x(\alpha), \alpha), M(\alpha)v(x(\alpha), \alpha) \rangle] d\alpha \quad (2)$$

where for all α : F , R and M are positive definite matrices and Q is a positive semi-definite matrix.

It is well known that the value function of this game, if it exists, is given by:

$$V(x, t) = 1/2 x' P(t) x \quad (3)$$

where (x, t) is any initial phase and $P(t)$ is the solution to the following matrix Riccati-type equation:

$$\dot{P}(t) + P(t)A(t) + A'(t)P(t) + P(t)[C(t)M^{-1}(t)C'(t) - B(t)R^{-1}(t)B'(t)]P(t) + Q(t) = 0; \quad (4) \\ P(T) = F.$$

Also, the optimal strategies are:

$$u^0(x, t) = -R^{-1}(t) B'(t) P(t) x \quad (5)$$

$$v^0(x, t) = M^{-1}(t) C'(t) P(t) x \quad (6)$$

Consider the game Riccati equation (4) with $Q(\cdot) \equiv 0$:

$$\dot{P}(t) + P(t)A(t) + A'(t)P(t) + P(t)[C(t)M^{-1}(t)C'(t) - B(t)R^{-1}(t)B'(t)]P(t) = 0; \quad (7) \\ P(T) = F.$$

Under the "relative controllability" assumption [2] the solution to this equation exists for $t \leq T$ and is unique and symmetric. The observation that setting $B(\cdot) = 0$ or $C(\cdot) = 0$ in (7) gives the Riccati equation for a minimum or a maximum control problem suggests that P for a game may be expressible in terms of P 's for control problems, i.e. the value function of the game might be related to the "value functions" of associated control problems. Such a relationship will now be demonstrated for two classes of differential games.

3. COMPOSITION PROBLEM-LINEAR PURSUIT-EVASION GAMES

For the linear pursuit evasion game with $Q(\cdot) = 0$, let $P(\cdot)$ be the solution to (7), and let $P_0(\cdot)$, $P_1(\cdot)$ and $P_2(\cdot)$ be such that for all $t \leq T$:

$$a. \quad \dot{P}_0(t) + P_0(t)A(t) + A'(t)P_0(t) = 0; \quad P_0(T) = F, \quad (8)$$

$$b. \quad \dot{P}_1(t) + P_1(t)A(t) + A'(t)P_1(t) - P_1(t)B(t)R^{-1}(t)B'(t)P_1(t) = 0; \quad (9) \\ P_1(T) = F.$$

$$c. \quad \dot{P}_2(t) + P_2(t)A(t) + A'(t)P_2(t) - P_2(t)C(t)M^{-1}(t)C'(t)P_2(t) = 0; \quad (10)$$

$$P_2(T) = F. \quad (10)$$

$$d. \quad P_2(t) - P_1(t) > 0 \quad (11)$$

Then:

$$1. \quad P(t) = [P_0^{-1}(t) + P_1^{-1}(t) - P_2^{-1}(t)]^{-1}, \quad t \leq T, \quad (12)$$

$$2. \quad P(t) > 0 \quad t \leq T \quad (13)$$

To prove this, note that since $Q(\cdot) \equiv 0$, P^{-1} satisfies the following linear differential equation:

$$\frac{d}{dt} P^{-1}(t) = A(t)P^{-1}(t) + P^{-1}(t)A'(t) + C(t)M^{-1}(t)C'(t) - B(t)R^{-1}(t)B'(t); \quad (14)$$

$$P^{-1}(T) = F^{-1}$$

Similarly:

$$\frac{d}{dt} P_0^{-1}(t) = A(t)P_0^{-1}(t) + P_0^{-1}(t)A'(t); \quad P_0^{-1}(T) = F^{-1} \quad (15)$$

$$\frac{d}{dt} P_1^{-1}(t) = A(t)P_1^{-1}(t) + P_1^{-1}(t)A'(t) - B(t)R^{-1}(t)B'(t); \quad P_1^{-1}(T) = F^{-1} \quad (16)$$

$$\frac{d}{dt} P_2^{-1}(t) = A(t)P_2^{-1}(t) + P_2^{-1}(t)A'(t) - C(t)M^{-1}(t)C'(t); \quad P_2^{-1}(T) = F^{-1} \quad (17)$$

It is easy to verify that $P_0^{-1} + P_1^{-1} - P_2^{-1}$ satisfies (14), and so $(P_0^{-1} + P_1^{-1} - P_2^{-1})^{-1}$ satisfies (7) provided the inverse exists for $t \leq T$. Results for optimal control theory [3,4] indicate that P_0 , P_1 and P_2 exist and are unique for $t \leq T$. In addition, $P_0 > 0$, $P_1 > 0$ and $P_2 > 0$. Together with (11) this implies that $P_1^{-1} - P_2^{-1} > 0$ [5] and that $P_0^{-1} + P_1^{-1} - P_2^{-1} > 0$. Therefore, $[P_0^{-1} + P_1^{-1} - P_2^{-1}]^{-1}$ exists for $t \leq T$.

It should be noted that $[P_0^{-1} + P_1^{-1} - P_2^{-1}]$ is not just a direct superposition of the homogeneous solution and the forced response of $CM^{-1}C'$ and $BR^{-1}B'$ in (14) since (17) is not obtained by simply setting $B \equiv 0$ in (14). A glance at (8) through (10) reveals that P_0 is the Riccati matrix for the "null control" problem, i.e. neither player exerts a control effort. In fact, (11) corresponds to a one-sided minimization problem defined by:

$$\dot{x} = A(t)x + C(t)v; \quad x(t_0) = x_0 \quad (18)$$

$$\text{and} \quad J_v = \frac{1}{2} \langle x(T), F x(T) \rangle + \frac{1}{2} \int_{t_0}^T \langle v(\alpha), M(\alpha)v(\alpha) \rangle d\alpha \quad (19)$$

A similar remark holds for P_1 and (9) so that the solution to the differential game can be obtained from the solutions of two state regulator control problems in which each player controls the same plant that is used in the game and attempts to drive the plant state to the origin while minimizing the fuel cost. Although in the game context the evader seeks to maximize J

while the pursuer simultaneously attempts to minimize J , their optimal performance in the game can be determined from their individual performances in a state regulator control problem. The term P_0 accounts for the autonomous motion of the plant; this same contribution to the payoff is present in P_1 and P_2 (due to the linearity of the plant and since $Q \equiv 0$) but is cancelled when we take $P_1^{-1} - P_2^{-1}$ to form P .

Had we chosen to consider the composition of P by letting $B \equiv 0$ in (14) assumption "d" would have to be replaced by one which would ensure the existence of the Riccati matrix P_2 for the corresponding maximization problem. This condition follows directly from the known conditions for the value. It is well known that the value for the game exists if and only if

$$\begin{aligned} & F^{-1} + \int_t^T \phi(T, \alpha) B(\alpha) R^{-1}(\alpha) B'(\alpha) \phi'(T, \alpha) d\alpha - \\ & \int_t^T \phi(T, \alpha) C(\alpha) M^{-1}(\alpha) C'(\alpha) \phi'(T, \alpha) d\alpha > 0 \end{aligned} \quad (20)$$

for every t in $[t_0, T]$. Hence, setting $B = 0$ gives the condition for the existence of P_2 . Our assumption that $P_2 - P_1 > 0$ guarantees that for any initial phase (x, t) the pursuer can attain a better control problem payoff than the evader. Due to the relationship between P_1 and P_2 and the controllability matrices for the pursuer and evader in (20) this condition is equivalent to assuming that the pursuer is more controllable than the evader, as pointed out by Ho, et.al. in [2]. This relationship, when observed with (20) could also prompt a search for a relationship between the Riccati matrices for the game and those for the control problems.

Finally, note that since a matrix Riccati equation of the form

$$\dot{P} + PA + A'P + PZP = 0; P(T) = F > 0, T \leq T, \quad (21)$$

has a linear equation for $P^{-1}(t)$ the solution $P(t)$ can be written as $P(t) = [P_0^{-1}(t) + P_n^{-1}(t) - P_m^{-1}(t)]^{-1}$ where:

$$\begin{aligned} \dot{P}_n + P_n A + A' P_n - P_n N(t) P_n &= 0; P_n(T) = F \\ \dot{P}_m + P_m A + A' P_m - P_m M(t) P_m &= 0; P_m(T) = F \\ \dot{P}_0 + P_0 A + A' P_0 &= 0; P_0(T) = F \end{aligned}$$

and $M \geq 0$, $N \geq 0$ and $M - N = Z = Z'$ and it is assumed that $P_m - P_n > 0$. Hence, given a Riccati equation (21) one can relate its solution to control problems and decompose its solution in arbitrarily many ways to correspond to differential games for a system

$$\dot{X} = A(t)X + H'_n u + H'_m v; x(t_0) = x_0$$

$$J = \langle x(T), F x(T) \rangle + \int_{t_0}^T [\langle u(\alpha), I_{P_n} u(\alpha) \rangle - \langle v(\alpha), I_{P_m} v(\alpha) \rangle] d\alpha$$

where $N = H'_n H_n$, $M = H'_m H_m$, p_m and p_n are the rank of M and N respectively, and I_{P_m} and I_{P_n} are identity matrices of corresponding order [6].

Likewise, there are many possible M and N which realize Z , corresponding to the fact that many system models and performance indices can be associated with a given solution to the matrix Riccati equation.

4. COMPOSITION PROBLEM - NORM-INVARIANT SYSTEMS

A conflict described by a norm-invariant system [7] and a time-to-capture payoff is a second example of the composition problem. Let:

$$\dot{x} = f(x, t) + bu(t) + cv(t); x(t_0) = x_0$$

where $x \in \mathbb{R}^n$; b and c are scalars; u and v are n -vector control functions for the pursuer and evader respectively, with $u, v: \mathbb{R}^n \rightarrow \mathbb{R}^n$. The range of u , A_1 , is given by $A_1 = \{\mu: \mu \in \mathbb{R}^n, \|\mu\| \leq 1\}$. Likewise, the range of v , A_2 , is:

$A_2 = \{\beta: \beta \in \mathbb{R}^n, \|\beta\| \leq 1\}$. The target set, S , is given by $S = \{(x, t): (x, t) \in \mathbb{R}^n \times \mathbb{R}, \|x\| = a > 0\}$. For a given control law pair (u, v) and initial phase (x_0, t_0) the payoff is given by the time to reach S and is represented by:

$$J(x_0, t_0, u, v) = \int_{t_0}^{t_1} d\alpha.$$

We must assume that $|b| > |c|$; otherwise E can effectively counter any choice made by P and prevent termination. Let $V(x, t)$ denote the time it takes for the phase to reach S when both players play optimally from the initial phase (x, t) ; the pursuer seeks to minimize V , the evader attempts to maximize V .

It has been shown [8] that

$$a) \quad V(x, t) = \frac{\|x\| - a}{|b| - |c|} \quad \text{for } \|x\| \geq a \quad (22)$$

$$b) \quad V(x, t) = \frac{a - \|x\|}{|b| - |c|} \quad \text{for } \|x\| \leq a \quad (23)$$

For our purposes it is sufficient to consider the case in which $\|x\| \geq a$. If we examine the control problems in which each player independently controls the plant and attempts to minimize the time to reach S , we would find:

$$V_1(x, t) = \frac{\|x\| - a}{|b|}, \quad (24)$$

$$V_2(x, t) = \frac{\|x\| - a}{|c|}, \quad (25)$$

where V_1 denotes the minimum time to reach S when the pursuer controls the plant, and V_2 denotes the minimum time when the evader controls the plant. Note that for $\|x\| > a$

$$\frac{1}{V(x, t)} = \frac{1}{V_1(x, t)} - \frac{1}{V_2(x, t)}.$$

In this case, the null control solution in which neither player exerts a control effort has a payoff $V_0(x, t) = \infty$ since the system is norm-invariant; i.e. $\langle f(x, t), x \rangle = 0$ for every (x, t) . Then, for $\|x\| > a$:

$$\frac{1}{V(x, t)} = \frac{1}{V_0(x, t)} + \frac{1}{V_1(x, t)} - \frac{1}{V_2(x, t)}. \quad (26)$$

The value functions in this example obey the same composition rule that the Riccati matrices obeyed in the linear pursuit-evasion example, where $V = x'P(t)x$ and $P(t) = [P^{-1} + P_1^{-1} - P_2^{-1}]^{-1}$.

To interpret the composition property for the norm-invariant game note

first that in each region of the space the pursuer uses the same control law that is used in the optimal control problem in the absence of the evader: $u^0(x,t) = -\frac{bx}{|b||x|}$ for $\|x\| > a$ and $u^0(x,t) = \frac{bx}{|b||x|}$ for $\|x\| < a$.

This is reasonable since the state feedback control law from the optimal control problem indicates the best choice of control for a given state, and the presence of the evader doesn't affect this choice because the performance index is the same in the game and in the corresponding control problems. However, in the linear pursuit-evasion case this did not hold.

For a given state x , $\|x\| > a$, the pursuer, acting alone, can reach the target in $\frac{\|x\| - a}{|b|}$ seconds, or, on an instantaneous basis travels to the

target at a "rate" of $|b|/[\|x\| - a]$. In the game the "rates" are subtracted to obtain the net rate of $[|b| - |c|]/[\|x\| - a]$.

It should now be more transparent why we chose to consider the composition of the Riccati matrix from control problem solutions rather than from a superposition of contributions to the game Riccati equation. Indeed, the (minimization) control problems can probably always be formulated to have meaning, while the maximization problem from the game may not have meaning - as is the case when one allows the evader to control the norm-invariant plant and maximize the time to reach S .

5. COMMENTS

It does not appear that the composition property lends itself to a reduction of the computational effort in solving the matrix Riccati equation for the game. However, it does provide additional insight to the nature of differential games and to the close relationship of control theory to differential games. We hope that a more complete theory of this relationship will eventually be developed and that the examples presented here will serve to stimulate interest in this area.

ACKNOWLEDGEMENT

The author wishes to express his gratitude for the motivation and guidance which he received from Professor Celso de Renna e Souza throughout the work which led to this paper.

The work in this paper was partially supported by NSF Grant GK 355 at the Department of Electrical Engineering, University of Notre Dame.

REFERENCES

1. Liu, R. W., and Leake, R. J., "Exhaustive Equivalence Classes of Optimal Systems with Separable Controls", J. SIAM Control, Vol. 4, 1966, pp. 678-685.
2. Ho, Y. C., Bryson, A. E., and Baron, S., "Differential Games and Optimal Pursuit-Evasion Strategies", IEEE Transactions on Automatic Control, Vol. AC-10, 1965, pp. 385-389.
3. Kalman, R. E., "Contributions to the Theory of Optimal Control", Bol. Soc. Mat. Mexicana, Second Ser., Vol. 5, 1960, pp. 102-119.
4. Athans, M., and Falb, P. L., "Optimal Control: An Introduction to the Theory and its Applications", McGraw-Hill, New York, 1966, pp. 750-766.
5. Bellman, R., Introduction to Matrix Analysis, McGraw-Hill, New York, 1960, p. 59.

6. Kalman, R. E., "When Is a Linear Control System Optimal?", J. Basic Engineering, Vol. 86, 1964, p. 54.
7. Athans, M., Falb, P., and Lacoss, R., "Time, Fuel, and Energy Optimal Control of Nonlinear Norm-Invariant Systems", IEEE Transactions Automatic Control, Vol. AC-8, 1963, pp. 196-202.
8. Ciletti, M. D., "Norm-Invariant Systems - Differential Games", to appear in IEEE Trans. Automatic Control, Vol. AC-13, No. 6.

WILLIAM S. LEVINE and MICHAEL ATHANS
Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

This paper considers the optimal control of linear, possibly time-varying, systems with respect to quadratic performance criteria. An additional constraint is imposed such that the control vector $\underline{u}(t)$ is a linear time-varying function of the output vector $\underline{y}(t)$ ($\underline{u}(t) = -\underline{F}(t)\underline{y}(t)$) rather than of the state vector $\underline{x}(t)$. An "averaged" optimization problem is formulated; the optimal feedback matrix $\underline{F}(t)$ is then specified by the necessary conditions provided by the matrix minimum principle.

1. INTRODUCTION

It is often the case that the state variables of a linear system are not available for feedback purposes; moreover, the designer may not wish to use a Kalman filter [1] or a state reconstructor [2] to obtain estimates of the state variables. Rather, one may wish to generate the control variables via linear feedback from the output variables; in this case, one is interested in determining an optimal set of feedback gains. This paper represents an effort to solve this class of problems.

In order to place the paper into perspective a more precise formulation is necessary. Consider a linear system whose state vector $\underline{x}(t)$ (an n -vector), control vector $\underline{u}(t)$ (an m -vector), and output vector $\underline{y}(t)$ (an r -vector) are related by

$$\dot{\underline{x}}(t) = \underline{A}(t)\underline{x}(t) + \underline{B}(t)\underline{u}(t) \quad (1.1)$$

$$\underline{y}(t) = \underline{C}(t)\underline{x}(t) \quad (1.2)$$

Consider also the standard quadratic cost functional

$$J = \frac{1}{2} \underline{x}'(T)\underline{S}\underline{x}(T) + \frac{1}{2} \int_{t_0}^T [\underline{x}'(t)\underline{Q}(t)\underline{x}(t) + \underline{u}'(t)\underline{R}(t)\underline{u}(t)] dt \quad (1.3)$$

It is well known [3] that the optimal control can be generated by $\underline{u}(t) = -\underline{G}(t)\underline{x}(t)$ where the gain matrix $\underline{G}(t)$ can be evaluated through the solution of the Riccati equation.

Now suppose that one introduces the constraints that the control $\underline{u}(t)$ be generated via output linear feedback, i.e.

$$\underline{u}(t) = -\underline{F}(t)\underline{y}(t) \quad (1.4)$$

or

$$\underline{u}(t) = -\underline{F}(t)\underline{C}(t)\underline{x}(t) \quad (1.5)$$

where $\underline{F}(t)$, the feedback gain matrix, is to be determined. Under such constraints, the system (1.1) is described by

*This research was conducted at the MIT Electronic Systems Laboratory with partial support provided by the NASA Electronics Research Center under grant NGL-22-009(124) and by the National Science Foundation under grant GK-2645.

$$\dot{\underline{x}}(t) = [\underline{A}(t) - \underline{B}(t)\underline{F}(t)\underline{C}(t)]\underline{x}(t) \quad (1.6)$$

Thus, the choice of the gain matrix $\underline{F}(t)$ obviously governs the closed loop dynamics of the system. In fact, the response of the closed loop system is

$$\underline{x}(t) = \underline{\Phi}(t, t_0)\underline{x}(t_0) \quad (1.7)$$

where $\underline{\Phi}(t, t_0)$ is the fundamental transition matrix, associated with $[\underline{A}(t) - \underline{B}(t)\underline{F}(t)\underline{C}(t)]$, defined by

$$\dot{\underline{\Phi}}(t, t_0) = [\underline{A}(t) - \underline{B}(t)\underline{F}(t)\underline{C}(t)]\underline{\Phi}(t, t_0) ; \underline{\Phi}(t_0, t_0) = \underline{I} \quad (1.8)$$

If we substitute Eqs. (1.5) and (1.7) into the cost functional (1.3) then we deduce that, for any given initial state $\underline{x}(t_0)$ and any given feedback matrix $\underline{F}(t)$, the cost is

$$J = \frac{1}{2} \underline{x}'(t_0) \left[\underline{\Phi}'(T, t_0) \underline{S} \underline{\Phi}(T, t_0) + \int_{t_0}^T \underline{\Phi}'(t, t_0) [\underline{Q}(t) + \underline{C}'(t)\underline{F}'(t)\underline{R}(t)\underline{F}(t)\underline{C}(t)] \underline{\Phi}(t, t_0) dt \right] \underline{x}(t_0) \quad (1.9)$$

or, equivalently,

$$J = \frac{1}{2} \{ \text{tr} [\underline{\Phi}'(T, t_0) \underline{S} \underline{\Phi}(T, t_0) \underline{x}(t_0) \underline{x}'(t_0)] + \frac{1}{2} \int_{t_0}^T \text{tr} \{ \underline{\Phi}'(t, t_0) [\underline{Q}(t) + \underline{C}'(t)\underline{F}'(t)\underline{R}(t)\underline{F}(t)\underline{C}(t)] \underline{\Phi}(t, t_0) \underline{x}(t_0) \underline{x}'(t_0) \} dt \} \quad (1.10)$$

Equation (1.10) brings into focus the dependence of the cost upon the initial state $\underline{x}(t_0)$ and the gain matrix $\underline{F}(t)$.

It is desired to find an "optimal" value for $\underline{F}(t)$ independent of the initial state. This requires that we give up "optimality" in return for "simplicity." To do this we attempt to determine $\underline{F}(t)$ which is optimal in an "average" sense (as was the case in refs. [4] and [5]). If we view the initial state $\underline{x}(t_0)$ as a random variable uniformly distributed over the surface of an n -dimensional unit sphere, then the expected value $\hat{J}$ of the cost (1.10) is simply

$$\hat{J} = \frac{1}{2} \text{tr} [\underline{\Phi}'(T, t_0) \underline{S} \underline{\Phi}(T, t_0)] + \frac{1}{2} \int_{t_0}^T \text{tr} [\underline{\Phi}'(t, t_0) [\underline{Q}(t) + \underline{C}'(t)\underline{F}'(t)\underline{R}(t)\underline{F}(t)\underline{C}(t)] \underline{\Phi}(t, t_0)] dt \quad (1.11)$$

The "average" cost $\hat{J}$ is now independent of the initial state $\underline{x}(t_0)$; it is, of course, dependent upon the choice of the gain matrix $\underline{F}(t)$. Therefore, it makes sense to seek a gain matrix $\underline{F}(t)$ which minimizes the average cost $\hat{J}$ subject to the differential constraint that the transition matrix $\underline{\Phi}(t, t_0)$ plays the role of the "state" and the matrix $\underline{F}(t)$ plays the role of the "control." Such problems can be readily attacked by the matrix minimum principle [6].

2. STATEMENT OF THE OPTIMIZATION PROBLEM

We consider the following optimization problem : Given the system described by the matrix differential equation

$$\dot{\underline{\Phi}}(t, t_0) = [\underline{A}(t) - \underline{B}(t)\underline{F}(t)\underline{C}(t)]\underline{\Phi}(t, t_0) ; \underline{\Phi}(t_0, t_0) = \underline{I} \quad (2.1)$$

and the cost functional

$$\hat{J} = \frac{1}{2} \int_{t_0}^T \text{tr} \{ \underline{\Phi}'(\tau, t_0) [\underline{Q}(\tau) + \underline{C}'(\tau) \underline{F}'(\tau) \underline{R}(\tau) \underline{F}(\tau) \underline{C}(\tau)] \underline{\Phi}(\tau, t_0) \} d\tau + \frac{1}{2} \text{tr} \{ \underline{\Phi}'(T, t_0) \underline{S} \underline{\Phi}(T, t_0) \} \quad (2.2)$$

Find the matrix $\underline{F}^*(t)$ that minimizes $\hat{J}$ subject to the differential constraints imposed by the system (2.1), where : $\underline{A}(t)$ is an $n \times n$ real matrix, $\underline{B}(t)$ is an $n \times m$ real matrix, $\underline{C}(t)$ is an $r \times n$ real matrix of full rank (rank r), $\underline{\Phi}(t, t_0)$ is an $n \times n$ matrix, $\underline{A}(t)$, $\underline{B}(t)$ and $\underline{C}(t)$ are locally integrable, $\underline{S}$ and $\underline{Q}(t)$ are $n \times n$ symmetric positive semi-definite real matrices, $\underline{R}(t)$ is an $m \times m$ symmetric positive definite real matrix, and $\underline{F}(t)$ is an $m \times r$ matrix.

$\underline{F}(t)$ is the "control" for the given system and is composed of measurable, but otherwise unconstrained, elements.

3. THE MAIN RESULT

The major result of this paper is summarized below. It specifies the properties of the optimal gain matrix $\underline{F}^*(t)$, $t \in [t_0, T]$.

The optimal gain matrix $\underline{F}^*(t)$, i.e., the one that minimizes the (average) cost (2.2) subject to the constraints (2.1) is given by

$$\underline{F}^*(t) = \underline{R}^{-1}(t) \underline{B}'(t) \underline{K}^*(t) \underline{\Phi}^*(t, t_0) \underline{\Phi}'(t, t_0) \underline{C}'(t) \underline{\Psi}^{-1}(t, t_0) \quad (3.1)$$

$$\text{where : (a) } \underline{\Psi}(t) \triangleq \underline{C}(t) \underline{\Phi}^*(t, t_0) \underline{\Phi}'(t, t_0) \underline{C}'(t) > 0 ; \underline{\Psi}(t) = \underline{\Psi}'(t) \quad (3.2)$$

(b) $\underline{\Phi}^*(t, t_0)$ is the solution of :

$$\dot{\underline{\Phi}}^*(t, t_0) = [\underline{A}(t) - \underline{B}(t) \underline{F}^*(t) \underline{C}(t)] \underline{\Phi}^*(t, t_0) ; \underline{\Phi}^*(t_0, t_0) = \underline{I} \quad (3.3)$$

(c) $\underline{K}^*(t)$ is the solution of :

$$\begin{aligned} \dot{\underline{K}}^*(t) = & -\underline{Q}(t) - \underline{C}'(t) \underline{F}^*(t) \underline{R}(t) \underline{F}^*(t) \underline{C}(t) - [\underline{A}(t) - \underline{B}(t) \underline{F}^*(t) \underline{C}(t)]' \underline{K}^*(t) \\ & - \underline{K}^*(t) [\underline{A}(t) - \underline{B}(t) \underline{F}^*(t) \underline{C}(t)] \end{aligned} \quad (3.4)$$

with the boundary condition at the terminal time T :

$$\underline{K}^*(T) = \underline{S} \quad (3.5)$$

(d) Furthermore, the optimal $\underline{F}^*(t)$ is unique and absolutely continuous provided it exists.

Remarks : A satisfactory proof for the existence of $\underline{F}^*(t)$ is not available as yet; it is the topic of current investigations. The proof of the "main result" proceeds as follows :

- We shall show that $\underline{F}^*(t)$ satisfies the necessary conditions for optimality using the matrix minimum principle [6], in Theorem 1.
- We indicate that $\underline{K}^*(t)$, $\underline{\Phi}^*(t, t_0)$ and $\underline{F}^*(t)$ are unique (provided that they exist);
- We shall demonstrate that the Hamilton-Jacobi sufficiency conditions hold (provided that the solution exists).

All these proofs are summarized in the following section.

4. PROOF OF THE MAIN RESULT

Theorem 1 The matrices $\underline{F}^*(t)$, $\underline{K}^*(t)$, $\underline{\Phi}^*(t, t_0)$, $t \in [t_0, T]$, defined by Eqs. (3.1) to (3.5) satisfy the necessary conditions for optimality provided by the matrix minimum principle.

Proof : Let $\underline{P}(t, t_0)$ be an $n \times n$ "costate" matrix associated with $\underline{\Phi}(t, t_0)$. Then the (scalar) Hamiltonian function H for the optimization problem is given by

$$H = \frac{1}{2} \text{tr} \{ \underline{\Phi}'(t, t_0) [\underline{Q}(t) + \underline{C}'(t) \underline{F}'(t) \underline{R}(t) \underline{F}(t) \underline{C}(t)] \underline{\Phi}(t, t_0) \} \\ + \text{tr} \{ [\underline{A}(t) - \underline{B}(t) \underline{F}(t) \underline{C}(t)] \underline{\Phi}(t, t_0) \underline{P}'(t, t_0) \} \quad (4.1)$$

We note that the Hamiltonian is a quadratic function of $\underline{F}(t)$. Hence, a necessary condition that $\underline{F}^*(t)$ minimizes H is that the following gradient matrix [6] vanishes:

$$\left. \frac{\partial H}{\partial \underline{F}} \right|_* = \underline{R}(t) \underline{F}^*(t) \underline{C}(t) \underline{\Phi}^*(t, t_0) \underline{\Phi}^{*'}(t, t_0) \underline{C}'(t) - \underline{B}'(t) \underline{P}^*(t, t_0) \underline{\Phi}^{*'}(t, t_0) \underline{C}'(t) = 0 \quad (4.2)$$

Hence $\underline{F}^*(t)$ is given by

$$\underline{F}^*(t) = \underline{R}^{-1}(t) \underline{B}'(t) \underline{P}^*(t, t_0) \underline{\Phi}^{*'}(t, t_0) \underline{C}'(t) \underline{\Psi}^{-1}(t) \quad (4.3)$$

$$\text{where : } \underline{\Psi}(t) \triangleq \underline{C}(t) \underline{\Phi}^*(t, t_0) \underline{\Phi}^{*'}(t, t_0) \underline{C}'(t) \quad (4.4)$$

Note that $\underline{\Psi}(t)$ is symmetric and at least positive semidefinite. Since $\underline{\Phi}(t, t_0)$ is a transition matrix and since $\underline{C}(t)$ is of rank r , then $\underline{\Psi}^{-1}(t)$ exists and, so, $\underline{\Psi}(t)$ is positive definite.

It is possible to establish, using tedious algebraic manipulation that $\underline{F}^*(t)$ does indeed minimize the Hamiltonian. This proof is, however omitted.

Using the necessary conditions of the matrix minimum principle one deduces that the "costate" matrix $\underline{P}^*(t, t_0)$ satisfies the matrix equation

$$\dot{\underline{P}}^*(t, t_0) = - \left. \frac{\partial H}{\partial \underline{\Phi}(t, t_0)} \right|_* \\ = - [\underline{Q}(t) + \underline{C}'(t) \underline{F}^*(t) \underline{R}(t) \underline{F}^*(t) \underline{C}(t)] \underline{\Phi}^*(t, t_0) - [\underline{A}(t) - \underline{B}(t) \underline{F}^*(t) \underline{C}(t)]' \underline{P}^*(t, t_0) \quad (4.5)$$

and, of course, $\underline{\Phi}^*(t, t_0)$ satisfies the equation

$$\dot{\underline{\Phi}}^*(t, t_0) = [\underline{A}(t) - \underline{B}(t) \underline{F}^*(t) \underline{C}(t)] \underline{\Phi}^*(t, t_0) ; \quad \underline{\Phi}^*(t_0, t_0) = \underline{I} \quad (4.6)$$

Furthermore, at the terminal time T

$$\underline{P}^*(T, t_0) = \left. \frac{\partial}{\partial \underline{\Phi}(T, t_0)} \text{tr} [\underline{\Phi}'(T, t_0) \underline{S} \underline{\Phi}(T, t_0)] \right|_* = \underline{S} \underline{\Phi}^*(T, t_0) \quad (4.7)$$

We claim that the solutions of (4.5) and (4.6) are related by

$$\underline{P}^*(t, t_0) = \underline{K}^*(t) \underline{\Phi}^*(t, t_0) \quad (4.8)$$

where $\underline{K}^*(t)$ is an $n \times n$ matrix to be determined. From Eq. (4.8)

$$\dot{\underline{P}}^*(t, t_0) = \dot{\underline{K}}^*(t) \underline{\Phi}^*(t, t_0) + \underline{K}^*(t) \dot{\underline{\Phi}}^*(t, t_0) \quad (4.9)$$

Substituting Eqs. (4.5) and (4.6) and (4.8) into Eq. (4.9) we obtain

$$- [\underline{Q}(t) + \underline{C}'(t) \underline{F}^*(t) \underline{R}(t) \underline{F}^*(t) \underline{C}(t)] \underline{\Phi}^*(t, t_0) - [\underline{A}(t) - \underline{B}(t) \underline{F}^*(t) \underline{C}(t)]' \underline{K}^*(t) \underline{\Phi}^*(t, t_0) \\ = \dot{\underline{K}}^*(t) \underline{\Phi}^*(t, t_0) + \underline{K}^*(t) [\underline{A}(t) - \underline{B}(t) \underline{F}^*(t) \underline{C}(t)] \underline{\Phi}^*(t, t_0) \quad (4.10)$$

which yields

$$\dot{\underline{K}}^*(t) = - \underline{Q}(t) - \underline{C}'(t) \underline{F}^*(t) \underline{R}(t) \underline{F}^*(t) \underline{C}(t) - \underline{K}^*(t) [\underline{A}(t) - \underline{B}(t) \underline{F}^*(t) \underline{C}(t)] \\ - [\underline{A}(t) - \underline{B}(t) \underline{F}^*(t) \underline{C}(t)]' \underline{K}^*(t) \quad (4.11)$$

From Eqs. (4.7) and (4.8) we deduce that

$$\underline{K}^*(T) = \underline{S} \quad (4.12)$$

Finally, from Eqs. (4.8) and (4.3) we have

$$\underline{F}^*(t) = \underline{R}^{-1}(t) \underline{B}'(t) \underline{K}^*(t) \underline{\Phi}^*(t, t_0) \underline{\Phi}^{*'}(t, t_0) \underline{C}'(t) \underline{\Psi}^{-1}(t) \quad (4.13)$$

This completes the proof that the matrices $\underline{F}^*(t)$, $\underline{\Phi}^*(t, t_0)$ and $\underline{K}^*(t)$,

as stated in Section 3, satisfy the necessary conditions for optimality.

We remark that both $K^*(t)$ and $\Phi^*(t, t_0)$ satisfy matrix differential equations. It can be shown that they satisfy (local) Lipschitz conditions; this implies that the solutions are (locally) unique.

We shall next prove that $\Phi^*(t, t_0)$ and $K^*(t)$ satisfy the Hamilton-Jacobi equation.

Theorem 2 : $F^*(t)$, $K^*(t)$ and $\Phi^*(t, t_0)$, defined by equations (3.1)-(3.5) satisfy the sufficiency conditions that result from the application of the Hamilton-Jacobi theorem.

Proof : Define

$$\underline{V}^*(t) = \int_t^T \Phi^{*-1}(t, t_0) \Phi^{*'}(\tau, t_0) [Q(\tau) + C'(\tau) F^*(\tau) R(\tau) F^*(\tau) C(\tau)] \Phi(t, \tau) \Phi^{*-1}(t, t_0) d\tau + \Phi^{*-1}(t, t_0) \Phi^{*'}(T, t_0) S \Phi(T, t_0) \Phi^{*-1}(t, t_0) \quad (4.14)$$

Differentiating with respect to time, we obtain :

$$\dot{\underline{V}}^*(t) = -Q(t) - C'(t) F^*(t) R(t) F^*(t) C(t) - \underline{V}^*(t) [A(t) - B(t) F^*(t) C(t)] - [A(t) - B(t) F^*(t) C(t)]' \underline{V}^*(t) \quad (4.15)$$

with terminal condition $\underline{V}^*(T) = S$.

Notice that $\underline{V}^*(t)$ exists and is unique because the above equation is linear. Notice also that $\underline{V}^*(t) = K^*(t)$ because they both satisfy the identical differential equation and boundary condition. We can evaluate the cost functional (2.2) to obtain

$$\hat{J}(F^*, t) = \frac{1}{2} \text{tr}[\Phi^{*'}(t, t_0) \underline{V}^*(t) \Phi^*(t, t_0)] \quad (4.16)$$

We can now compute the derivatives of $\hat{J}$:

$$\frac{\partial \hat{J}(F^*, t, \Phi^*)}{\partial t} = \frac{1}{2} \text{tr}[\Phi^{*'}(t, t_0) \dot{\underline{V}}^*(t) \Phi^*(t, t_0)] \quad (4.17)$$

$$\frac{\partial \hat{J}(F^*, t, \Phi^*)}{\partial \Phi^*(t, t_0)} = \underline{V}^*(t) \Phi^*(t, t_0) = K^*(t) \Phi^*(t, t_0) = P^*(t, t_0) \quad (4.18)$$

$$\text{Let : } H(\Phi^*(t, t_0), F^*(t), t, \frac{\partial \hat{J}(F^*, t, \Phi^*)}{\partial \Phi^*(t, t_0)}) \triangleq H^* \quad (4.19)$$

$$\text{But : } H^* = \frac{1}{2} \text{tr}[\Phi^{*'}(t, t_0) \{Q(t) + C'(t) F^*(t) R(t) F^*(t) C(t)\} \Phi^*(t, t_0)] + \text{tr}[\{A(t) - B(t) F^*(t) C(t)\} \Phi^*(t, t_0) \Phi^{*'}(t, t_0) \underline{V}^*(t)] \quad (4.20)$$

$$\text{or : } H^* = \frac{1}{2} \text{tr}\{\Phi^{*'}(t, t_0) [Q(t) + C'(t) F^*(t) R(t) F^*(t) C(t)] \Phi^*(t, t_0)\} + \frac{1}{2} \text{tr}\{\Phi^{*'}(t, t_0) [A(t) - B(t) F^*(t) C(t)]' \Phi^*(t, t_0)\} \quad (4.21)$$

$$\text{or : } H^* = \frac{1}{2} \text{tr}\{\Phi^{*'}(t, t_0) [Q + C' F^* R F^* C + \underline{V}^* (A - B F^* C) + (A - B F^* C)' \underline{V}^*] \Phi^*(t, t_0)\} \quad (4.22)$$

$$\text{Hence : } H^* = \frac{1}{2} \text{tr}\{\Phi^{*'}(t, t_0) [-\dot{\underline{V}}^*(t)] \Phi^*(t, t_0)\} \quad (4.23)$$

This implies that

$$\frac{\partial \hat{J}(F^*, t, \Phi^*)}{\partial t} + H^* = 0 \quad (4.24)$$

In other words, the Hamilton-Jacobi equation is satisfied, as is the terminal condition.

Thus, we have verified all the assumptions of the Hamilton-Jacobi theorem [as given by theorem 5.13, p. 360, of reference [3]].

5. COMPUTATION OF $F^*(t)$

In order to implement the closed loop system one must be able to compute the feedback gain matrix $F^*(t)$, $t \in [t_0, T]$. The equations that specify $F^*(t)$ are given in section 2. It should be obvious that the specification of $F^*(t)$ involves the solution of a nonlinear two-point boundary value problem involving matrix differential equations; very little work has been done in this area.

We outline below a computational procedure which generates a sequence of matrices $\{K_n(t)\}$, $\{F_n(t)\}$, and $\{\phi_n(t, t_0)\}$ which, hopefully, converge to the optimal ones. We prove that this algorithm has the property that the cost decreases at each iteration.

The algorithm for computing $K_{n+1}(t)$, $F_{n+1}(t)$ and $\phi_{n+1}(t, t_0)$ is based on the stored value of

$$F_n(t) = R^{-1}(t)B'(t)K_n(t)\phi_n(t, t_0)\phi_n'(t, t_0)C'(t)[C(t)\phi_n(t, t_0)\phi_n'(t, t_0)C'(t)]^{-1} \quad (5.1)$$

Thus, to start the algorithm we require an initial guess $F_0(t)$. We obtain this guess by setting

$$K_0(t) = S \text{ and } \phi_0(t, t_0) = I \text{ in Eq. (5.1).} \quad (5.2)$$

Knowing $F_n(t)$, we compute $K_{n+1}(t)$ by integrating the equation

$$\begin{aligned} \dot{K}_{n+1}(t) = & -K_{n+1}(t)[A(t)-B(t)F_n(t)C'(t)] - [A(t)-B(t)F_n(t)C(t)]'K_{n+1}(t) - Q(t) \\ & - C'(t)F_n(t)R(t)F_n(t)C(t) \end{aligned} \quad (5.3)$$

backwards in time from the terminal condition $K_{n+1}(T) = S$. Knowing $K_{n+1}(t)$, we then substitute into equation (5.1) to obtain

$$\begin{aligned} F_{n+1}(t) = & R^{-1}(t)B'(t)K_{n+1}(t)\phi_{n+1}(t, t_0)\phi_{n+1}'(t, t_0)C'(t) \\ & [C(t)\phi_{n+1}(t, t_0)\phi_{n+1}'(t, t_0)C'(t)]^{-1} \end{aligned} \quad (5.4)$$

Note that the value of $F_{n+1}(t)$ is not known as yet since $\phi_{n+1}(t, t_0)$ has not been evaluated. The expression for $F_{n+1}(t)$ is substituted into Eq. (5.5) below to obtain a nonlinear differential equation for $\phi_{n+1}(t, t_0)$. This equation is numerically integrated forward in time to obtain $\phi_{n+1}(t, t_0)$, $t \in [t_0, T]$, and simultaneously $F_{n+1}(t)$ via Eq. (5.4). Now all three matrices $K_{n+1}(t)$, $F_{n+1}(t)$, and $\phi_{n+1}(t, t_0)$ have been evaluated, $F_{n+1}(t)$ has been stored, and the algorithm continues.

$$\dot{\phi}_{n+1}(t, t_0) = [A(t)-B(t)F_{n+1}(t)C'(t)]\phi_{n+1}(t, t_0) ; \phi_{n+1}(t_0, t_0) = I \quad (5.5)$$

In section 4 we proved that the minimum cost

$$\hat{J}(F^*, t) = \frac{1}{2} \text{tr}[\Phi^*(t, t_0)K^*(t)\Phi^*(t, t_0)] \quad (5.6)$$

so that $\hat{J}(F^*, t_0) = \frac{1}{2} \text{tr}[K^*(t_0)]$. Then, it can be shown that $\hat{J}(F_n, t_0) = \frac{1}{2} \text{tr}[K_{n+1}(t_0)]$, which equals the cost when $F_n(t)$ is used as the feedback gain matrix.

Lemma - Using the algorithm described above,

$$\text{tr}[K_{n+1}(t_0)] \leq \text{tr}[K_n(t_0)] \quad \text{for all } n \quad (5.7)$$

Proof -

$$\begin{aligned} \frac{d}{dt} [K_n(t) - K_{n+1}(t)] = & -[A - BF_{n-1}C]'K_n - K_n[A - BF_{n-1}C] - C'F_{n-1}R F_{n-1}C \\ & + [A - BF_nC]'K_{n+1} + K_{n+1}[A - BF_nC] + C'F_n R F_n C \end{aligned} \quad (5.8)$$

$$\begin{aligned} \frac{d}{dt} [K_n(t) - K_{n+1}(t)] = & -[A - BF_n C]' K_n + [A - BF_n C]' K_{n+1} - K_n [A - BF_n C] + K_{n+1} [A - BF_n C] \\ & + (BF_{n-1} C)' K_n - (BF_n C)' K_n + K_n (BF_{n-1} C) - K_n BF_n C - C' F_{n-1} R R^{-1} R F_n C \\ & + C' F_n R R^{-1} R F_n C + K_n B R^{-1} B' K_n - K_n B R^{-1} B' K_n \end{aligned} \quad (5.9)$$

$$\begin{aligned} = & -[A - BF_n C]' (K_n - K_{n+1}) - (K_n - K_{n+1}) [A - BF_n C] \\ & - [C' F_{n-1} R - K_n B] R^{-1} [R F_{n-1} C - B' K_n] + [C' F_n R - K_n B] R^{-1} [R F_n C - B' K_n] \end{aligned} \quad (5.10)$$

Integrating Eq. (5.10) and taking the trace, we obtain :

$$\begin{aligned} \text{tr}[K_{n+1}(t_0)] - \text{tr}[K_n(t_0)] = & \int_{t_0}^T \text{tr}\{\phi_n'(\tau, t_0) [(C' F_{n-1} R - K_n B) R^{-1} (R F_{n-1} C - B' K_n) \\ & - (C' F_n R - K_n B) R^{-1} (R F_n C - B' K_n)] \phi_n(\tau, t_0)\} d\tau \end{aligned} \quad (5.11)$$

We now wish to show that the integrand in equation (5.11) is positive for all τ . To do this, we first introduce three facts.

I) If $\underline{X}$ is a real symmetric $n \times n$ matrix, $\text{tr}[\underline{X}] = \sum_{i=1}^n \underline{x}_i' \underline{X} \underline{x}_i$ where $\{\underline{x}_i\}$ is an arbitrary orthonormal basis for R^n . (5.12)

Proof - $\underline{x}_i = \underline{P} \underline{e}_i$ where $\underline{e}_i$ is an element of the natural basis and $\underline{P}$ is an orthogonal matrix ($\underline{P} \underline{P}' = \underline{I}$). Then $\text{tr}[\underline{X}] = \text{tr}[\underline{P}' \underline{X} \underline{P}] = \sum_{i=1}^n \underline{e}_i' \underline{P}' \underline{X} \underline{P} \underline{e}_i = \sum_{i=1}^n \underline{x}_i' \underline{X} \underline{x}_i$

II) $[\underline{C}(\tau) \phi_n(\tau, t_0) \phi_n'(\tau, t_0) \underline{C}'(\tau)]^{-1} \underline{C}(\tau) \phi_n(\tau, t_0) = [\phi_n'(\tau, t_0) \underline{C}'(\tau)]^\dagger$ (5.13)

where $\dagger$ denotes pseudo-inverse.

Proof - See reference [7], p. 582.

III) If $\underline{x}_0$ is an arbitrary real n -vector, $\underline{x}_0$ can be written as

$$\begin{aligned} \underline{x}_0 = \underline{x}_1 + \underline{x}_2 \text{ where } \underline{x}_1 \in \mathcal{R}[\phi_n'(\tau, t_0) \underline{C}'(\tau)] \\ \underline{x}_2 \in \mathcal{R}[\phi_n'(\tau, t_0) \underline{C}'(\tau)]^\perp = \mathcal{N}[\underline{C}(\tau) \phi_n(\tau, t_0)] \end{aligned}$$

In order to evaluate the trace under the integral at time τ , we choose an orthonormal basis $\{\underline{x}_i\}$ such that $\underline{x}_i \in \mathcal{R}[\phi_n'(\tau, t_0) \underline{C}'(\tau)]$ for $i \leq m$ (5.14)

$$\underline{x}_i \in \mathcal{R}[\phi_n'(\tau, t_0) \underline{C}'(\tau)]^\perp \text{ for } m < i \leq n$$

Using facts I and III, we see that the integrand in equation (5.11)

$$\begin{aligned} \text{tr}\{\phi_n'(\tau, t_0) [(C' F_{n-1} R - K_n B) R^{-1} (R F_{n-1} C - B' K_n) \\ - (C' F_n R - K_n B) R^{-1} (R F_n C - B' K_n)] \phi_n(\tau, t_0)\} \stackrel{\Delta}{=} \text{tr}[\underline{I}_n] \end{aligned} \quad (5.15)$$

$$\text{tr}[\underline{I}_n] = \sum_{i=1}^n \underline{x}_i' \underline{I}_n \underline{x}_i \text{ where } \underline{x}_i \text{ are elements of the special basis (5.14).} \quad (5.16)$$

We see that for $\underline{x}_i \in \mathcal{N}[\underline{C}(\tau) \phi_n(\tau, t_0)]$ or equivalently, $\underline{x}_i$ such that $m < i \leq n$:

$$\underline{x}_i' \underline{I}_n \underline{x}_i = \underline{x}_i' \phi_n'(\tau, t_0) [K_n B R^{-1} B' K_n - K_n B R^{-1} B' K_n] \phi_n(\tau, t_0) \underline{x}_i = 0 \quad (5.17)$$

For all the other $\underline{x}_i$, $\underline{x}_i \in \mathcal{R}[\phi_n'(\tau, t_0) \underline{C}'(\tau)]$, we make use of fact II to show,

$$\begin{aligned} \underline{F}_n(\tau) \underline{C}(\tau) \phi_n(\tau, t_0) = \underline{R}^{-1}(\tau) \underline{B}'(\tau) \underline{K}_n(\tau) \phi_n(\tau, t_0) \phi_n'(\tau, t_0) \underline{C}'(\tau) \\ [\underline{C}(\tau) \phi_n(\tau, t_0) \phi_n'(\tau, t_0) \underline{C}'(\tau)]^{-1} \underline{C}(\tau) \phi_n(\tau, t_0) \end{aligned} \quad (5.18)$$

or,

$$\underline{F}_n(\tau)\underline{C}(\tau)\underline{\Phi}_n(\tau, t_0) = \underline{R}^{-1}(\tau)\underline{B}'(\tau)\underline{K}_n(\tau)\underline{\Phi}_n(\tau, t_0) + [\underline{\Phi}_n'(\tau, t_0)\underline{C}'(\tau)][\underline{\Phi}_n'(\tau, t_0)\underline{C}'(\tau)]^+ \quad (5.19)$$

Using theorem 5, p. 579, of ref. 7.

$$\underline{F}_n(\tau)\underline{C}(\tau)\underline{\Phi}_n(\tau, t_0)\underline{x}_1 = \underline{R}^{-1}(\tau)\underline{B}'(\tau)\underline{K}_n(\tau)\underline{\Phi}_n(\tau, t_0)\underline{x}_1 \quad \text{for } \underline{x}_1 \in \mathcal{R}[\underline{\Phi}_n'\underline{C}'] \quad (5.20)$$

Thus, for the remaining $\underline{x}_1$, $1 \leq m$,

$$\underline{x}_1' \underline{I}_{n-n} \underline{x}_1 = \underline{x}_1' \underline{\Phi}_n'(\tau, t_0) [(\underline{C}' \underline{F}'_{n-1} \underline{R} - \underline{K}' \underline{B}) \underline{R}^{-1} (\underline{R} \underline{F}_{n-1} \underline{C} - \underline{B}' \underline{K}_n)] \underline{\Phi}_n(\tau, t_0) \underline{x}_1 \geq 0 \quad (5.21)$$

Hence, the integrand in equation (5.10) is positive and the lemma is proven.

6. EXAMPLE

The following example is included primarily to illustrate the results presented in this paper. It is a purely academic example and was not chosen to dramatically prove the utility of this approach. In fact, since the open loop transfer function contains an unstable pole and since the system is not stabilizable with respect to constant output feedback, the example is a relatively poor one for using output feedback.

Given the linear system :

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 4 & -4 & 1 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t) \quad (6.1)$$

$$y(t) = \underline{x}_1(t) \quad ; \quad \text{with } u(t) = F(t)y(t) \quad ; \quad t \in [0, T] \quad (6.2)$$

$$G(s) = \frac{1}{(s-1)(s^2+4)} \quad (6.3)$$

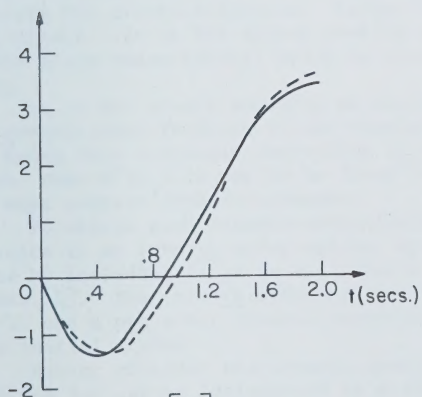
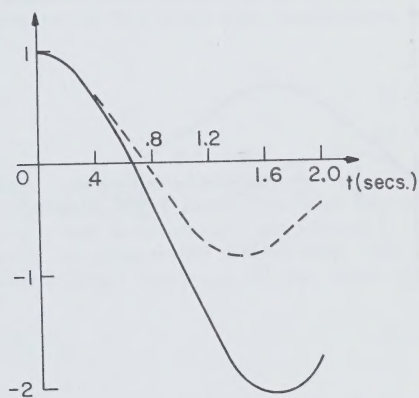
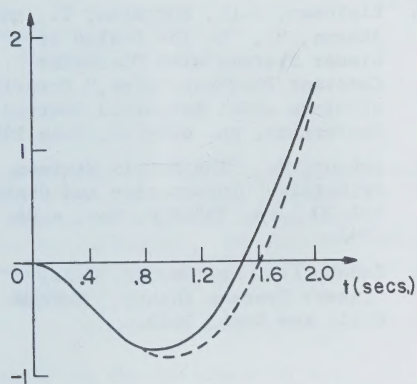
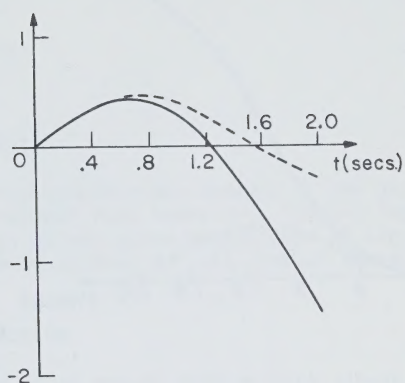
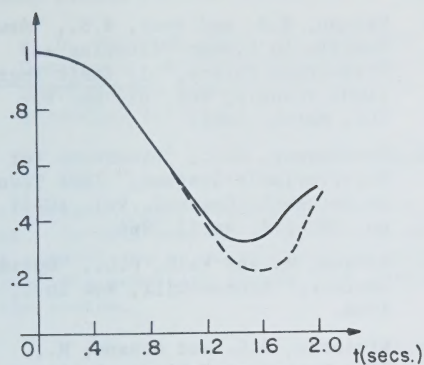
and the cost criterion :

$$J = \frac{1}{2} \int_0^{T^2} [20y^2(t) + u^2(t)] dt \quad (6.4)$$

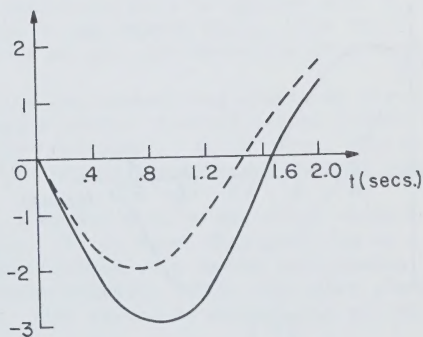
$$J = \frac{1}{2} \int_0^{T^2} \text{tr}[20y^2(t) + y'(t)F^2(t)y(t)] dt \quad (6.5)$$

We wish to choose $F(t)$ to minimize the cost criterion (6.5).

The state trajectories of the optimal output feedback controlled system are plotted as the solid curves on the following pages. For comparison, the state trajectories of the optimal state feedback controlled system are plotted as dashed lines on the same graphs. It should be noted that, using performance criterion (6.5) the output feedback system performs about 55% worse than the state feedback system. However, as is obvious from the graphs, there are large differences in performance. It should be noted that this ease in comparing performance for different available information is a major advantage of this technique.



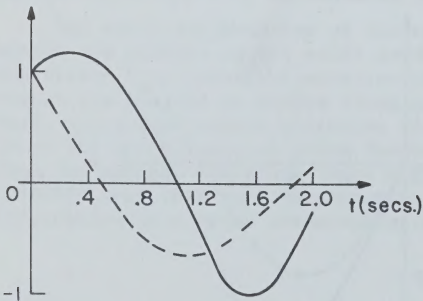
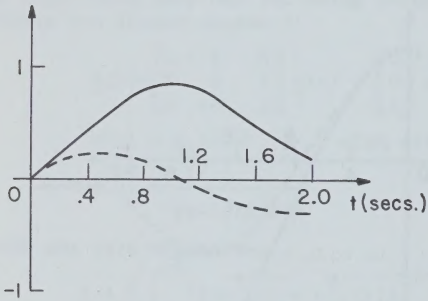
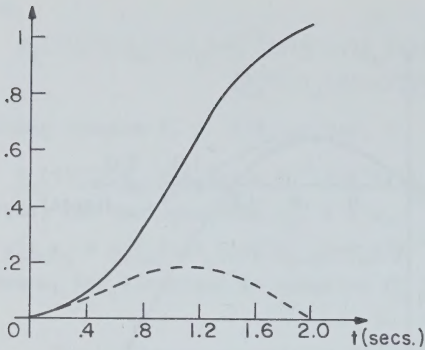
$$\underline{x}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$



$$\underline{x}_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

References

1. Kalman, R.E. and Bucy, R.S., "New Results in Linear Filtering and Prediction Theory," J. Basic Engrg., (ASME Trans.), Vol. 83, pp. 95-108, March, 1961.
2. Luenberger, D.G., "Observers for Multivariable Systems," IEEE Trans. on Automatic Control, Vol. AC-11, pp. 190-197, April 1966.
3. Athans, M. and Falb, P.L., "Optimal Control," McGraw-Hill, New York, 1966.
4. Kleinman, D.L. and Athans, M., "The Design of Suboptimal Linear Time-Varying Systems," IEEE Trans. on Automatic Control, Vol. AC-13, pp. 150-159, April 1968.
5. Kleinman, D.L., Fortmann, T., and Athans, M., "On the Design of Linear Systems with Piecewise Constant Feedback Gains," Preprints of ninth Joint Automatic Control Conference, pp. 698-710, June 1968.
6. Athans, M., "The Matrix Minimum Principle," Information and Control, Vol. 11, pp. 592-606, Nov, - Dec. 1967.
7. Zadeh, L.A. and Desoer, C.A., "Linear Systems Theory," McGraw-Hill, New York, 1963.



$$\underline{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

OPTIMAL LINEAR CONTROL SYSTEMS WITH INCOMPLETE STATE MEASUREMENTS

J. D. FERGUSON AND Z. V. REKASIUS
Department of Electrical Engineering
Northwestern University
Evanston, Illinois

ABSTRACT

Optimal linear regulators require the complete measurement of the plant state. In this paper a procedure is developed that leads to a dynamic feedback control law which explicitly depends on any given measurement of the state. The resulting closed loop system is optimal for all initial states of the system.

INTRODUCTION

Optimal control laws, such as those which result from solving linear regulator problems, require the measurement of each state variable of the plant. However, in practice one cannot measure each state variable of a plant. This represents a serious restriction on the practical usefulness of optimal control theory.

Consider the plant

$$\dot{x} = Ax + bu \quad (1-1)$$

$$y_1 = C_1 x \quad (1-2)$$

where A is an $n \times n$ matrix, b is an n vector, and C_1 is a $m \times n$ matrix, $m \leq n$, $\text{rank } C_1 = m$. A feedback control law $U^*(y_1)$ is desired such that the resulting closed loop system is optimal (i.e., so that a quadratic performance index is minimized). For the case where C_1 is square and invertible, the solution is known and has the form of proportional feedback of the state of the plant.

$$u^*(t) = U^*(y_1(t)) = -k'x = -k'C_1^{-1}y_1 \quad (1-3)$$

where k' is the transpose of k .

Also, Kalman [1] has solved the inverse problem of determining when a linear system is optimal. The system must be asymptotically stable and satisfy the circle criterion. Kalman's result can also be applied when C_1 is not square. It is the direct problem with C_1 not square (i.e., with incomplete state measurements) which is unsolved as yet, and therefore of interest here.

It is not always possible to stabilize the closed loop system by means of proportional feedback of the measurements (1-2). From classical theory it is known that a dynamic controller is often necessary. It is clear, therefore, that $U^*(y_1)$ is not to be found in the set of proportional control laws; it must contain dynamic elements.

To obtain such dynamic controllers, a method will be described which results in an $(n+p)$ th order optimal system. It is known that this can be done by including terms in the first p derivatives of u in the performance index [2]. The $(n+p)$ th order optimal system consists of the n th order plant (1-1) and a p th order dynamic controller. The system is optimal for all of its initial states.

Having obtained the dynamic control law, it will be shown that the control law can be implemented as a function of the given set of measurements (1-2), provided $p \geq v-1$, where v is the observability index of the plant. In general, $n/m \leq v \leq n-m+1$.

In this way, the desired control law is obtained for the case where C_1

is not square. The result implies that a linear compensator can be designed to close the loop around a linear plant so that the resulting system is optimal.

OPTIMAL DYNAMIC CONTROLLER

Consider again the plant (1-1), (1-2) and the performance index

$$I = \int_{t_0}^{\infty} (x'Q_1'Q_1x + x'Q_2'Q_2z + \beta(u^{(k)})^2) d\tau \quad (2-1)$$

where

$$z' = (u, u, \dots, u^{(k-1)}), \beta > 0, \text{ and } u^{(i)} = \frac{d^i u}{dt^i}.$$

By definition $z_i = z_{i+1}$, $i = 1, \dots, k-1$.

With the designation $u^{(k)} = w$, the equations can be written

$$\dot{x} = Ax + bu \quad (2-2(a))$$

$$y_1 = C_1 x$$

$$\dot{z} = Dz + ew \quad (2-2(b))$$

$$u = f'z$$

where the $[D, e]$ pair is in phase variable form with zero eigenvalues and $f' = (1, 0, \dots, 0)$. Then w is selected to minimize (2-1). If the composite plant (2-2) is completely controllable, one obtains the optimal control law

$$w^* = W^*(x, z) = -(g'x + h'z) \quad (2-3)$$

INCOMPLETE STATE FEEDBACK

The optimal system (2-2), (2-3) implies the complete feedback of both x and z . However, while the plant (2-2(a)) is fixed, there is freedom regarding the implementation of the controller (2-2(b)), (2-3). To eliminate the feedback from the state variables not measured, consider the transformation

$$\bar{z} = z + Tx \quad (3-1)$$

The system (2-2), (2-3) becomes

$$\dot{x} = Ax + bu \quad (3-2)$$

$$y_1 = C_1 x$$

$$\dot{\bar{z}} = (D - eh')\bar{z} + Kx + Tbu$$

$$u = f'\bar{z} + k_0'x$$

where $K = (TA - (D - eh')T - eg')$

$$k_0' = -f'T$$

Since the system (2-2), (2-3) and the system (3-2) are related by the transformation (3-1), they are equivalent mathematical representations of the same system. But, with reference to implementation, in (3-2) the feedback from x is through $k_0'x$ and Kx . Thus, feedback is introduced to the controller at points other than the k th derivative of u . It is this assumed freedom of a multi-input controller that will make possible the elimination of feedback from unavailable state variables.

One wants to find T so that $k_0'x$ and Kx are functions only of the measurements available (1-2). This implies

$$\begin{bmatrix} K \\ k'_0 \end{bmatrix} x = \begin{bmatrix} R \\ r'_0 \end{bmatrix} y_1 = \begin{bmatrix} R \\ r'_0 \end{bmatrix} C_1 x \quad (3-3)$$

for all x and for some R and r_0 . Thus it is required that

$$\begin{bmatrix} TA-(D-eh')T-eg' \\ -f'T \end{bmatrix} = \begin{bmatrix} R \\ r'_0 \end{bmatrix} C_1 \quad (3-4)$$

for some T , R and r_0 . The situation with regard to the solvability of (3-4) is summarized in the following theorem.

Theorem 1: If $[A, C_1]$ and $[(D-eh'), f']$ are both completely observable then (3-4) always has a solution, provided the order of the controller is at least $\nu-1$, where ν is the observability index of $[A, C_1]$.

Theorem 1 is proven in Appendix I.

ν is the minimum integer so that the matrix Q has rank n , where

$$Q = [C_1', A'C_1', \dots, A'^{\nu-1}C_1']$$

In general, $n/m \leq \nu \leq n-m+1$.

Using T , R and r_0 which solve (3-4), one obtains the desired optimal system with incomplete state feedback as follows.

$$\begin{aligned} \dot{x} &= Ax + bu \\ y_1 &= C_1 x \\ \dot{z} &= (D-eh')z + Ry_1 + Tbu \\ u &= f'z + r'_0 y_1 \end{aligned} \quad (3-5)$$

A GENERALIZATION

The development of a dynamic controller in section 2 is somewhat restrictive. In general, consider the plant (1-1) and the performance index

$$I = \int_{t_0}^{\infty} (x'Q_1'Q_1x + z'Q_2'Q_2z + \beta w^2) dt \quad (4-1)$$

with $\beta > 0$. Instead of using u as the control, adjoin to (1-1) an arbitrary dynamic controller

$$\begin{aligned} \dot{z} &= Dz + ew \\ u &= f'z \end{aligned} \quad (4-2)$$

Then make w the control. If the composite (x, z) plant is completely controllable, one obtains the optimal solution

$$w^* = W^*(x, z) = -(g'x + h'z) \quad (4-3)$$

Thus, the controller (4-2) is of arbitrary form, with arbitrary eigenvalues. Since the development in section 3 is carried out for a general controller, no changes are necessary there, and the incomplete state feedback solution can be obtained with this more general controller.

The choice of the controller (4-2) is arbitrary (except for its order and proper controllability properties). The eigenvalues are completely arbitrary. The choice of (4-2) should be motivated by practical design considerations. For example, the designer could have in mind several "standard" first order compensators which would be interconnected to form the controller (4-2). In this context, it does not make sense to think of "the" optimal solution. The closed loop solution depends upon the choice of (4-2).

TIME VARYING PROBLEMS

The results described in sections 1 to 4 can be extended to time varying problems. By a development analogous to the stationary case, one obtains the following condition that must be satisfied (analogous to (3-4)).

$$\begin{aligned} \dot{T}(t) + T(t)A(t) - (D(t) - e(t)h'(t))T(t) - e(t)g'(t) &= R(t) \\ -f'(t)T(t) &= r_0'(t) \quad C_1(t) \end{aligned} \quad (5-1)$$

for some $T(t)$, $R(t)$ and $r(t)$.

If (5-1) has a solution, one obtains the desired optimal system with incomplete state feedback as follows.

$$\begin{aligned} \dot{x} &= A(t)x + b(t)u \\ y_1 &= C_1(t)x \\ \dot{z} &= (D(t) - e(t)h'(t))\bar{z} + R(t)y_1 + T(t)b(t)u \\ u &= f'(t)\bar{z} + r_0'(t)y_1 \end{aligned} \quad (5-2)$$

Theorem 2: If $[A, C_1]$ and $[(D - eh'), f']$ are both uniformly observable, then (5-1) always has a solution, provided the order of the controller is at least $\nu - 1$, where ν is the observability index of $[A, C_1]$.

A proof of Theorem 2 is given in [3]. The general argument is the same as in the proof of Theorem 1, which is given in Appendix I.

The pair $[A, C_1]$ is uniformly observable if the matrix Q has rank n for all t , where

$$Q = (P_0, P_1, \dots, P_{r-1}), \text{ for some } r \leq n \text{ and } P_{k+1} = A'P_k + \frac{d}{dt}P_k,$$

$P = C'$. ν is the minimum value of r .

Example: Consider the second order plant

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -a_1(t) & -a_2(t) \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (5-3)$$

Adjoin the controller

$$\begin{aligned} \dot{z} &= w \\ u &= z \end{aligned}$$

Choose w to minimize the performance index

$$I = \int_{t_0}^T (x_1^2 + x_2^2 + z^2 + w^2) d\tau \quad (5-4)$$

for all $x(t_0)$ and $z(t_0)$, and T fixed.

The optimal control law, found by solving the third order matrix Riccati equation, can be written

$$w^* = \bar{W}^*(x, z, t) = -(g'(t)x + h'(t)z) \quad (5-5)$$

T , R and r_0 are of the following form.

$$T = (T_1(t), T_2(t)), \quad R = R_1(t), r_0' = R_0(t)$$

Then (5-1) can be written for this problem as follows:

$$\dot{T}_1 - a_1(t)T_2 + h(t)T_1 - g_1(t) = R_1(t) \quad (5-6)$$

$$\dot{T}_2 + T_1 - a_2(t)T_2 + h(t)T_2 - g_2(t) = 0$$

$$-T_1 = R_0(t)$$

$$T_2 = 0$$

(5-6) is solved by

$$T = (g_2(t), 0)$$

$$R = \dot{g}_2(t) + h(t)g_2(t) - g_1(t)$$

$$r_0 = -g_2(t)$$

Thus, the optimal system with incomplete state feedback is

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -a_1(t) & -a_2(t) \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (5-7)$$

$$y_1 = [1 \ 0]x$$

$$\dot{z} = -hz + (\dot{g}_2 + hg_2 - g_1)y_1$$

$$u = \bar{z} - g_2y_1$$

CONCLUSION

The problem of incomplete state feedback has received much attention. It is suggested in [1] that the unavailable state variables be estimated with a Wiener filter. Then the controller would be the optimal estimator followed by the optimal proportional control (1-3). However, it is not clear how to generate the statistics required for a problem that is originally deterministic. Also, the complexity of the controller may be undesirable. From classical control theory it is known that satisfactory results can be obtained with relatively simple dynamic controllers.

The appeal of less complex estimators has motivated several designs. In [4,5] procedures are given for designing filters which give estimates of the optimal proportional control. These filters have order p where $n/m - 1 \leq p \leq n-m$. However, these filters are not designed for optimality.

In [6] optimality is redefined in terms of a min-max problem and a proportional incomplete state feedback solution is obtained. The method fails when the plant cannot be stabilized with proportional feedback of the available state variables. Further, the attractive properties of the optimal solution, as described in [1], are lost in the redefinition.

In contrast, the closed loop system described here is optimal in the sense of minimizing a quadratic performance index. The circle criterion of [1] is satisfied. This represents a significant improvement over the approaches used in [4,5,6]. Also, the dynamic order of the controller is the same as the filters designed in [4,5].

In a recent communication with the authors [7], Pearson reports similar results. Pearson considers the stationary case where the controller (2-2(b)) has zero eigenvalues.

REFERENCES

- 1) R. E. Kalman; "When is a Linear Control System Optimal?"; Transactions ASME, Series D, Journal of Basic Engineering, Vol. 86, pp. 51-60; March, 1964.

- 2) W. M. Wonham; "Variational Formulation of Control Problems"; CISL, School of Electrical Engineering, Purdue University, Lafayette, Indiana, unpublished lecture notes.
- 3) J. D. Ferguson; "Optimal Linear Control Systems with Incomplete State Measurements"; Ph.D. Dissertation, Department of Electrical Engineering, Northwestern University, Evanston, Illinois; June, 1969.
- 4) D. G. Luenberger; "Observing the State of a Linear System"; IEEE Transactions on Military Electronics, Vol. MIL-8, pp. 74-80; April, 1964.
- 5) D. G. Luenberger; "Observers for Multivariable Systems"; IEEE Transactions on Automatic Control, Vol. AC-11, pp. 190-197; April, 1966.
- 6) Z. V. Rekasius; "Optimal Linear Regulators with Incomplete State Feedback"; IEEE Transactions on Automatic Control, Vol. AC-12, pp. 296-299; June, 1967.
- 7) J. B. Pearson; "Compensator Design For Dynamic Optimization"; Department of Electrical Engineering, Rice University, Houston, Texas, unpublished.

Appendix I: Proof of Theorem 1

It is desired to find T , K , k_0 , R and r_0 so that the following equations hold:

$$K = TA - DT - eg' \quad (I-1)$$

$$k_0' = -f'T \quad (I-2)$$

$$K = RC_1 \quad (I-3)$$

$$k_0' - r_0' C_1 \quad (I-4)$$

where D in (I-1) is understood to be a shorthand writing of $(D - eh')$.

Define $\bar{K} = K + eg'$. Then (I-1) can be written as follows.

$$DT = -\bar{K} + TA \quad (I-5)$$

Premultiplying (I-5) by f' , one obtains

$$f'DT = -f'\bar{K} + f'TA \quad (I-6)$$

Substituting into (I-6) from (I-2), one obtains

$$f'DT = -f'\bar{K} - k_0'A \quad (I-7)$$

Premultiplying (I-5) by $f'D$, one obtains

$$f'D^2T = -f'D\bar{K} + f'DTA \quad (I-8)$$

Substituting into (I-8) from (I-7), one obtains

$$f'D^2T = -f'D\bar{K} - f'KA - k_0'A^2 \quad (I-9)$$

Continue in this fashion, premultiplying (I-5) by $f'D^{i-1}$, $i = 1, \dots, v-2$, thereby generating $v-2$ equations. The last equation generated is as follows.

$$\begin{aligned} f'D^{v-2}T = & -f'D^{v-3}\bar{K} - f'D^{v-4}\bar{K}A - \dots \\ & \dots - f'D\bar{K}A^{v-4} - f'\bar{K}A^{v-3} - k_0'A^{v-2} \end{aligned} \quad (I-10)$$

Let

$$Q_D = \begin{bmatrix} f' \\ f'D \\ \vdots \\ f'D^{v-2} \end{bmatrix} \quad (I-11)$$

Let

$$\bar{K}_1 = Q_D \bar{K} = \begin{bmatrix} \bar{k}'_1 \\ \bar{k}'_2 \\ \vdots \\ \bar{k}'_{v-1} \end{bmatrix} \quad (I-12)$$

With this notation, the $v-2$ equations generated before, taken together with (I-2), can be written as follows.

$$Q_D T = \begin{bmatrix} -k'_0 \\ -\bar{k}'_1 - k'_0 A \\ -\bar{k}'_2 - \bar{k}'_1 A - k'_0 A^2 \\ \vdots \\ -\bar{k}'_{v-2} - \bar{k}'_{v-3} A - \dots - k'_0 A^{v-2} \end{bmatrix} = S(\bar{K}_1, k_0) \quad (I-13)$$

Since $[D, f']$ is assumed completely observable, with order $v-1$, Q_D has an inverse. Then

$$T = Q_D^{-1} S(\bar{K}_1, k_0) \quad (I-14)$$

(I-14) represents a necessary condition. It is sufficient to show that (I-14) does satisfy (I-1) and (I-2) for K and k_0 given by (I-3) and (I-4) for some R and r_0 . (I-2) is satisfied identically. Substituting (I-14) into (I-5) one obtains

$$DQ_D^{-1} S(\bar{K}_1, k_0) = -\bar{K} + Q_D^{-1} S(\bar{K}_1, k_0) A \quad (I-15)$$

Premultiplying by Q_D and rearranging terms, one obtains

$$-S(\bar{K}_1, k_0) A + Q_D DQ_D^{-1} S(\bar{K}_1, k_0) = -Q_D \bar{K} = -\bar{K}_1 \quad (I-16)$$

By definition of Q_D , it follows that

$$Q_D DQ_D^{-1} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & 1 \\ f'D^{v-1} & & & & Q_D^{-1} \end{bmatrix} \quad (I-17)$$

Substituting from (I-17) and (I-13) into (I-16), one obtains

$$\begin{bmatrix} k'_0 A - \bar{k}'_1 - k'_0 A \\ \bar{k}'_1 A + k'_0 A^2 - \bar{k}'_2 - \bar{k}'_1 A - k'_0 A^2 \\ \vdots \\ \bar{k}'_{v-2} A + \bar{k}'_{v-3} A^2 + \dots + k'_0 A^{v-1} + f'D^{v-1} Q_D^{-1} S(\bar{K}_1, k_0) \end{bmatrix} = \begin{bmatrix} -\bar{k}'_1 \\ -\bar{k}'_2 \\ \vdots \\ -\bar{k}'_{v-1} \end{bmatrix} \quad (I-18)$$

The first $v-2$ rows of (I-18) are identities. The last row may be rewritten as follows.

$$\bar{k}'_{v-1} + \bar{k}'_{v-2} A + \bar{k}'_{v-3} A^2 + \dots + k'_0 A^{v-1} + f'D^{v-1} Q_D^{-1} S(\bar{K}_1, k_0) = 0 \quad (I-19)$$

Let

$$f'D^{\nu-1}Q_D^{-1} = d' = (d_1, d_2, \dots, d_{\nu-1}) \quad (I-20)$$

Also let $\bar{k}_0 = k_0$. Substituting from (I-13) and (I-20) into (I-19), transposing and rearranging terms, one obtains the following requirement.

$$\begin{aligned} F(\bar{K}_1, \bar{k}_0) = & \bar{k}_{\nu-1} + \sum_{i=0}^{\nu-2} \bar{k}_i d_{i+1} + A'(\bar{k}_{\nu-2} + \sum_{i=0}^{\nu-3} \bar{k}_i d_{i+2} + \dots \\ & \dots + A'^{\nu-2}(\bar{k}_1 + \bar{k}_0 d_{\nu-1}) + A'^{\nu-1} \bar{k}_0 = 0 \end{aligned} \quad (I-21)$$

Note that (I-12) defines $F(\bar{K}_1, k_0)$.

Define

$$K_1 = Q_D K \quad (I-22)$$

so that

$$\bar{K}_1 = Q_D \bar{K} = Q_D K + Q_D \text{eg}' = K_1 + Q_D \text{eg}'$$

Then, since F is linear in $\bar{K}_1$ and $\bar{k}_0$, (I-21) can be written as follows.

$$F(K_1, k_0) = -F(Q_D \text{eg}', 0) \quad (I-23)$$

It is at this point that the restrictions (I-3) and (I-4) are introduced. Premultiplying (I-3) by Q_D , one obtains

$$Q_D K = Q_D R C_1 = R_1 C_1 = \begin{bmatrix} r'_1 \\ r'_2 \\ \vdots \\ r'_{\nu-1} \end{bmatrix} \quad C_1 = K_1 = \begin{bmatrix} k'_1 \\ k'_2 \\ \vdots \\ k'_{\nu-1} \end{bmatrix} \quad (I-24)$$

Also, from (I-4), $k'_0 = r'_0 C_1$. Thus,

$$k'_i = C'_1 r'_i, \quad i = 0, \dots, \nu-1. \quad (I-25)$$

Substituting from (I-25) into (I-23), one obtains

$$\begin{aligned} C'_1(r'_{\nu-1} + \sum_{i=0}^{\nu-2} r'_i d_{i+1}) + A' C'_1(r'_{\nu-2} + \sum_{i=0}^{\nu-3} r'_i d_{i+2}) + \dots \\ \dots + A'^{\nu-2} C'_1(r'_1 + r'_0 d_{\nu-1}) + A'^{\nu-1} C'_1 r'_0 = -F(Q_D \text{eg}', 0) \end{aligned} \quad (I-26)$$

Define

$$q_j = r'_{\nu-1-j} + \sum_{i=0}^{\nu-2-j} r'_i d_{i+j+1}, \quad j = 0, \dots, \nu-2. \quad (I-27)$$

and

$$q_{\nu-1} = r'_0 \quad (I-28)$$

(I-27) and (I-28) can be written as follows.

$$[q_0, \dots, q_{\nu-1}] = [r'_0, \dots, r'_{\nu-1}] \begin{bmatrix} d_1 & d_2 & \dots & d_{\nu-1} & 1 \\ d_2 & d_3 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{\nu-1} & 1 & \dots & \vdots & \vdots \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix} \quad (I-29)$$

Substituting from (I-27) and (I-28) into (I-26), one obtains

$$[C_1', A' C_1', \dots, A'^{\nu-1} C_1'] \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{\nu-1} \end{bmatrix} = -F(Q_D \text{eg}', 0) \quad (\text{I-30})$$

If ν is the observability index of $[A, C_1]$, then the observability matrix on the left hand side of (I-30) has rank n and (I-30) has a solution. By substituting back through (I-29) and (I-24), one obtains the desired feedback matrices R and r_0 .

THE HUMAN OPERATOR AS AN OPTIMAL CONTROLLER

GARY L. BLANK
Litton Systems Inc.
Woodland Hills, California

PETER J. SCHUMACHER, JR.
Marquette University
Milwaukee, Wisconsin

ABSTRACT

Modern control theory is used to synthesize an optimal controller for a tracking task. The tracking problem is first converted to a regulator problem by the formulation of an augmented state vector and the judicious definition of the observation vector. These obviate the need for an adjoint system. Minimization of a practical quadratic cost functional results in the minimization of a Hamiltonian. This yields the optimal control law and associated canonical equations in terms of the state and costate vectors. Because of the linear relationship between the state and costate, a Riccati matrix differential equation is generated. The transversality condition provides the boundary condition for solving the Riccati equation on a digital computer.

A family of solutions is obtained for the Riccati equation for different values of weighting factors in the cost functional. These optimal controllers are simulated on an analog computer.

The optimal controller is replaced by the human operator who responds in a manner similar to the optimal controllers but almost identical to a particular one. From comparison of the simulation data a transfer function is obtained for the human operator on the premise that the human operator functions as an optimal controller with the prescribed cost functional in a tracking task.

INTRODUCTION

In recent years there has been increasing interest in the identification of the dynamics of the human operator in a tracking situation. Several of the more important methods used to characterize the operator are quasi-linearization (Reference 6), non-linear theory and computer simulations (Reference 4), sampled-data theory (Reference 2), and parallel filters (Reference 5).

In this paper, the identification of the human operator dynamics is performed in a unique manner. The procedure is motivated by the theory that the human operator may possibly be an optimal controller which minimizes a quadratic cost functional. If this theory can be substantiated, then the dynamics of the theoretically-derived optimal controller would match those of the human operator. The identification is then accomplished.

The optimal control law has been derived (Reference 1) for a tracker wherein a plant described by

$$\begin{aligned}\dot{\underline{x}} &= \underline{A} \underline{x} + \underline{B} \underline{u} \\ \underline{y} &= \underline{C} \underline{x}\end{aligned}\tag{1}$$

is required to track a system described by

$$\dot{\underline{z}} = \underline{D} \underline{z}\tag{2}$$

The cost functional to be minimized is

$$J = \frac{1}{2} \langle \underline{e}(T), \underline{F} \underline{e}(T) \rangle + \frac{1}{2} \int_0^T [\langle \underline{e}, \underline{Q} \underline{e} \rangle + \langle \underline{u}, \underline{R} \underline{u} \rangle] dt \quad (3)$$

where $\underline{e} = \underline{z} - \underline{y}$.

Using the variational calculus, the optimal controller yields a co-state vector which is related to the state vector by

$$\underline{p} = \underline{K} \underline{x} - \underline{g} \quad (4)$$

where $\underline{g}$ is the solution of a differential equation involving $\underline{z}$. This necessitates an adjoint system and knowledge of future values of $\underline{z}$. This result is generally not practical and an alternate formulation must be sought.

DEVELOPMENT OF THE OPTIMAL REGULATOR

For the tracking system described by Equations (1) and (2) an observation vector $\underline{y}$ may be constructed as follows:

$$\underline{y} = \underline{C} \underline{x}_a \quad (5)$$

$$\text{where } \underline{x}_a = \begin{bmatrix} \underline{z} \\ -\underline{x} \end{bmatrix} \quad (6)$$

The overall system equations may be written as

$$\begin{bmatrix} \dot{\underline{z}} \\ \dot{\underline{x}} \end{bmatrix} = \begin{bmatrix} \underline{D} & 0 \\ 0 & \underline{A} \end{bmatrix} \begin{bmatrix} \underline{z} \\ \underline{x} \end{bmatrix} + \begin{bmatrix} \underline{B}_1 \\ \underline{B}_2 \end{bmatrix} \underline{u} \quad (7)$$

$$\text{or } \dot{\underline{x}}_a = \underline{A}_1 \underline{x}_a + \underline{B} \underline{u} \quad (8)$$

$$\text{where } \underline{B} = \begin{bmatrix} \underline{B}_1 \\ -\underline{B}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (9)$$

$$\text{Let } \underline{C} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad (10)$$

so that the difference between $\underline{z}$ and $\underline{x}$ is observed. The system cost functional is given by

$$J = \frac{1}{2} \int_0^T [\langle \underline{y}, \underline{Q} \underline{y} \rangle + \langle \underline{u}, \underline{R} \underline{u} \rangle] dt \quad (11)$$

$$\text{where } \underline{Q} = \begin{bmatrix} q_{11} & 0 \\ 0 & q_{22} \end{bmatrix} \quad (12)$$

$$\text{and } \underline{R} = [r] \quad (13)$$

The two parts of this cost functional are penalty functions. The term $[\underline{y}, \underline{Q} \underline{y}]$ penalizes the system for values of $\underline{y}(t)$ as weighted by q_{ij} during the control interval $[0, T]$. $\langle \underline{u}, \underline{R} \underline{u} \rangle$ penalizes the system for the control level achieved during the control interval.

By the judicious selection of the elements of the $\underline{Q}$ and $\underline{R}$ matrices, the system response can be controlled as desired by the designer.

Upon minimizing the cost functional subject to the restraint of the system equation, the following Hamiltonian results:

$$H = \frac{1}{2} \langle \underline{y}, \underline{Q} \underline{y} \rangle + \frac{1}{2} \langle \underline{u}, \underline{R} \underline{u} \rangle + \langle \underline{A}_1 \underline{x}_a, \underline{p} \rangle + \langle \underline{B} \underline{u}, \underline{p} \rangle \quad (14)$$

where $\underline{p}$ is the costate vector of the system.

The optimal control $\underline{u}^*$ can be determined by minimizing H with respect to the control effort. Thus,

$$\frac{\partial H}{\partial \underline{u}} = \underline{R} \underline{u} + \underline{B}^T \underline{p} = 0 \quad (15)$$

from which

$$\underline{u}^* = -\underline{R}^{-1} \underline{B}^T \underline{p} \quad (16)$$

The remaining two canonical equations are

$$\frac{\partial H}{\partial \underline{x}_a} = -\dot{\underline{p}} = \underline{C}^T \underline{Q} \underline{C} \underline{x}_a + \underline{A}_1^T \underline{p} \quad (17)$$

$$\frac{\partial H}{\partial \underline{p}} = \dot{\underline{x}}_a = \underline{A}_1 \underline{x}_a - \underline{B} \underline{R}^{-1} \underline{B}^T \underline{p} \quad (18)$$

It has been proven (Reference 1) that $\underline{p}$ and $\underline{x}_a$ are linearly related by the equation

$$\underline{p} = \underline{K} \underline{x}_a \quad (19)$$

where $\underline{K}$ is an $n \times n$ symmetric time-varying matrix.

Equations (17), (18), and (19), can be combined and manipulated, resulting in a Riccati matrix equation

$$-\dot{\underline{K}} = \underline{K} \underline{A}_1 + \underline{A}_1^T \underline{K} - \underline{K} \underline{B} \underline{R}^{-1} \underline{B}^T \underline{K} + \underline{C}^T \underline{Q} \underline{C} \quad (20)$$

where $\underline{K}$ and $\dot{\underline{K}}$ are 4×4 symmetric time-varying matrices.

In order to solve Equation (20) for $\underline{K}$, one boundary condition is required. This is provided by the transversality condition which states that

$$\underline{p}(T) = \frac{\partial}{\partial \underline{x}(T)} \left[\frac{1}{2} \langle \underline{e}(T), \underline{F} \underline{e}(T) \rangle \right] \quad (21)$$

when the cost functional is

$$J = \frac{1}{2} \left\{ \langle \underline{x}(T), \underline{F} \underline{x}(T) \rangle + \int_0^T \langle \underline{x}, \underline{Q} \underline{x} \rangle + \langle \underline{u}, \underline{R} \underline{u} \rangle dt \right\} \quad (22)$$

for a system described by Equations (1) and (2).

The elements f_{ij} (of $\underline{F}$) in this case are weights assigned to the final values of the state variables x_i . $\langle \underline{x}(T), \underline{F} \underline{x}(T) \rangle$ penalizes the system for errors in the final value of $\underline{x}$.

Since the final value of the error is not considered important in a human tracking situation, the corresponding cost functional of Equation (11) excludes the $\langle \underline{x}(T), \underline{F} \underline{x}(T) \rangle$ term; consequently, $\underline{p}(T) = 0$. Furthermore, since $\underline{x}_a(T)$ is non-zero, from Equation (19) it follows that $\underline{K}(T) = 0$. This boundary condition can be used in solving for $\underline{K}$.

Combining Equations (16), (18), and (19) yields

$$\dot{\underline{x}}_a = \underline{A}_1 \underline{x}_a - \underline{B} \underline{R}^{-1} \underline{B}^T \underline{K} \underline{x}_a \quad (23)$$

The block diagram for the system is shown in Figure 1.

TRANSFER FUNCTION FOR THE OPTIMAL CONTROLLER

A model controller based upon the gains from the Riccati equation will now be formulated.

$$\text{Let } \underline{M} = \underline{B} \underline{R}^{-1} \underline{B}^T \hat{\underline{K}} \quad (24)$$

where $\hat{\underline{K}}$ is the solution to the Riccati equation based on infinite operating time.

Thus,

$$\underline{M} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\hat{k}_{14}}{r} & \frac{\hat{k}_{24}}{r} & \frac{\hat{k}_{34}}{r} & \frac{\hat{k}_{44}}{r} \end{bmatrix} \quad (25)$$

Partitioning $\underline{M}$ into four 2×2 matrices results in

$$\underline{M} = \begin{bmatrix} \underline{M}_{11} & \underline{M}_{12} \\ \underline{M}_{21} & \underline{M}_{22} \end{bmatrix} \quad (26)$$

$$\text{where } \underline{M}_{11} = \underline{M}_{12} = 0$$

$$\underline{M}_{21} = \begin{bmatrix} 0 & 0 \\ \frac{\hat{k}_{14}}{r} & \frac{\hat{k}_{24}}{r} \end{bmatrix} \quad \underline{M}_{22} = \begin{bmatrix} 0 & 0 \\ \frac{\hat{k}_{34}}{r} & \frac{\hat{k}_{44}}{r} \end{bmatrix} \quad (27)$$

and $\hat{k}_{ij}$ are the elements of $\hat{\underline{K}}$.

Let $\underline{C}$ also be partitioned into two 2×2 matrices

$$\underline{C} = \begin{bmatrix} \underline{C}_{11} & \underline{C}_{12} \end{bmatrix} \quad (28)$$

where

$$\underline{C}_{11} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \underline{C}_{12} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad (29)$$

The vector representation of the relationship between the observation $\underline{y}$ and $\underline{z}$ and $\underline{x}$ is shown in Figure 2. In this optimal regulator mechanization the reference trajectory is generated by the section with output $\underline{z}$. The controlled plant is composed of $\underline{M}_{21}$ and the section having the output $\underline{x}$.

The corresponding transfer function block diagram for the optimal tracker is shown in Figure 3. The transfer function for the system is

$$\frac{\underline{X}}{\underline{Z}}(s) = \frac{\frac{\hat{k}_{24}}{r} \left(s + \frac{\hat{k}_{14}}{\hat{k}_{24}} \right)}{s^2 + \left(a + \frac{\hat{k}_{44}}{r} \right) s + \frac{\hat{k}_{34}}{r}} \quad (30)$$

It is interesting to note that the control $\underline{u}$ is generated by difference signals obtained by weighing the desired signal $\underline{z}$ and comparing with the weighted outputs $\underline{x}$ from the plant.

If an error is defined as the difference between these weighted signals, that is,

$$e_1 = \frac{\hat{k}_{14}}{r} z_1 - \frac{\hat{k}_{34}}{r} x_1 \quad (31)$$

$$e_2 = \frac{\hat{k}_{24}}{r} z_2 - \frac{\hat{k}_{44}}{r} x_2 \quad (32)$$

then a transfer function for the optimal controller is obtained as

$$G(s) = \frac{U(s)}{E(s)} = \frac{\frac{\hat{k}_{24}}{r} \left(s + \frac{\hat{k}_{14}}{\hat{k}_{24}} \right) s (s + a)}{s^2 + s \left(a + \frac{\hat{k}_{44}}{r} - \frac{\hat{k}_{24}}{r} \right) + \frac{\hat{k}_{34}}{r} - \frac{\hat{k}_{14}}{r}} \quad (33)$$

SOLUTION TO THE RICCATI EQUATION

The Riccati matrix differential equation to be solved was derived above.

$$\dot{\underline{K}} = -\underline{K} \underline{A}_1 - \underline{A}_1^T \underline{K} + \underline{K} \underline{B} \underline{R}^{-1} \underline{B}^T \underline{K} - \underline{C}^T \underline{Q} \underline{C} \quad (34)$$

The boundary condition was shown to be $\underline{K}(T) = 0$. A method (Reference 1) for solving for $\underline{K}$ requires that Equation (34) be integrated backwards in time until the elements of $\underline{K}$ become constant, $\hat{\underline{K}}$. The

form of the integration is given by

$$\underline{K}(T + \Delta) = \underline{K}(T) + \Delta \{ \dot{\underline{K}}(T) \} \quad (35)$$

where Δ is a small negative increment of time

$\underline{K}(T)$ is the value of $\underline{K}$ which was the result of the previous iteration.

$\dot{\underline{K}}(T)$ is the value of $\dot{\underline{K}}$ of the previous iteration.

Steady state values for $\hat{k}_{14}$, $\hat{k}_{24}$, $\hat{k}_{34}$, and $\hat{k}_{44}$ were determined with the values of q_{11} and q_{22} independently chosen from the values 1, 10, and 100, and r was set equal to 1, .1, and .01 for each combination. The results for the 27 combinations are summarized in Table 1.

Inspection of the solutions to the Riccati equation indicates that

$$\frac{\hat{k}_{34}}{r} = \frac{\hat{k}_{14}}{r} \quad (36)$$

and

$$\left(a + \frac{\hat{k}_{44}}{r} - \frac{\hat{k}_{24}}{r} \right) \approx 0 \quad (37)$$

where the time constant of the plant $\frac{1}{a}$ was chosen to be 0.5 second.

Consequently, the transfer function derived for the optimal controller becomes from Equations (33), (36), and (37)

$$G(s) = \frac{\frac{\hat{k}_{24}}{r} \left(s + \frac{\hat{k}_{14}}{\hat{k}_{24}} \right) (s + a)}{s} \quad (38)$$

EXPERIMENTAL RESULTS OF THE HUMAN TRACKER

After the entire system had been simulated on a TR-10 analog computer using the gains developed above, the optimal controller was removed and a human operator inserted into the system. The operator was given a visual display of $e(t)$ and $\dot{e}(t)$ and was allowed to track $z(t)$ by means of a control stick as shown in Figure 4.

Strip chart recordings of the responses of the optimal controller and the human operator show remarkable similarities. This was especially true when the k_{ij} , r , and q_{ij} were set at the values shown as combination 16 in Table 1. For this set of values,

$$G(s) = \frac{10.3 (s + .3) (s + 2)}{s} \quad (39)$$

Equation (39) provides the transfer function of the human operator in this tracking task.

#	q_{11}	q_{22}	r	$\hat{k}_{14}$	$\hat{k}_{24}$	$\hat{k}_{34}$	$\hat{k}_{44}$
1	1	1	1	.96	2.16	1.00	.65
2	1	1	.1	.31	.43	.32	.25
3	1	1	.01	.10	.11	.10	.09
4	1	10	1	.96	3.54	1.00	2.00
5	1	10	.1	.31	1.03	.32	.85
6	1	10	.01	.10	.32	.10	.30
7	1	100	1	.96	9.93	.99	8.30
8	1	100	.1	.31	3.16	.32	2.98
9	1	100	.01	.10	.98	.10	.96
10	10	1	1	3.14	3.14	3.16	1.36
11	10	1	.1	1.00	.57	1.00	.38
12	10	1	.01	.32	.13	.32	.11
13	10	10	1	3.14	4.28	3.16	2.51
14	10	10	.1	1.00	1.10	1.00	.91
15	10	10	.01	.32	.33	.32	.31
16	10	100	1	3.14	10.29	3.16	8.50
17	10	100	.1	.99	3.19	1.00	3.00
18	10	100	.01	.31	.98	.31	.96
19	100	1	1	9.99	4.86	10.00	3.00
20	100	1	.1	3.16	.87	3.16	.68
21	100	1	.01	1.00	.17	1.00	.15
22	100	10	1	9.99	5.69	10.00	3.83
23	100	10	.1	3.16	1.28	3.16	1.09
24	100	10	.01	1.00	.35	1.00	.33
25	100	100	1	9.99	11.0	10.00	9.14
26	100	100	.1	3.16	3.26	3.16	3.07
27	100	100	.01	0.98	.99	.98	.97

STEADY STATE VALUES FOR $\hat{k}_{ij}$ OF RICCATI EQUATION
TABLE 1

It is interesting to compare this transfer function with that obtained by McRuer and Krendel (Reference 6) in evaluating a pilot in a simulated aircraft environment with a random signal input. For a signal bandwidth of 1 rad/second,

$$G(s) = \frac{100e^{-0.15s} (2s + 1)}{(25s + 1) (.67s + 1)} \quad (40)$$

Realizing that the input signal for the system described in this paper was deterministic with a natural frequency of 0.1 rad/second, a reasonable approximation to Equation (40) yields

$$G(s) \doteq \frac{8(s + 0.5)}{s} \quad (41)$$

Note the similarity between Equations (39) and (41). The zero at -2 in Equation (39) is due to the fact that the optimal controller (operator) was able to learn the task and effectively cancel the pole of the plant at -2, whereas, in Equation (41) the operator was unable to learn and predict because of the random nature of the signal.

CONCLUSION

Optimal control theory has been used in a unique application - the characterization of the human operator. It was theorized that the human operator possibly judges his own performance in a tracking task with respect to a quadratic performance index. Synthesis of an optimal controller for causing a specific plant to track led to the solution of a Riccati matrix differential equation. Discrete arbitrary weighting factors were used in the cost functional resulting in a family of optimal controllers. The human operator was observed to respond in a manner similar to one of these controllers leading to the conjecture that the human operator dynamics were similar to those of the controller.

The results do not conclusively prove that the operator uses the specific performance index described in this paper. However, the technique described and the results have established a new approach to the identification of operator dynamics with the use of modern optimal control theory.

ACKNOWLEDGMENT

The authors wish to thank B. Kumar for her invaluable help in the preparation of the manuscript.

REFERENCES

1. Athans, M. and Falb, P., "Optimal Control", McGraw-Hill Book Co., Inc., N. Y., 1966.
2. Bekey, G. A., "The Human Operator as a Sampled Data System", I. R. E. Transactions on Human Factors in Electronics, HFE - 3(2), 1962.
3. McRuer, D. T., Krendel, E. S., "The Human Operator as a Servo System", Journal of the Franklin Institute, Vol. 269, Nos. 5 and 6, 1959.
4. Young, L. R., Meiry, J. F., "Bang-Bang Aspects of Manual Control in High-Order Systems", IEEE Transactions on Automatic Control, Vol. AC-10, No. 6, July 1965.
5. Krendel, E. S., McRuer, D. T., "A Servomechanism Approach to Skill Development", Journal of the Franklin Institute, Vol. 269, No. 1, 1960.
6. McRuer, D. T., Krendel, E. S., "Dynamic Response of Human Operators", WADC Tech. Report 56-524, 1957.

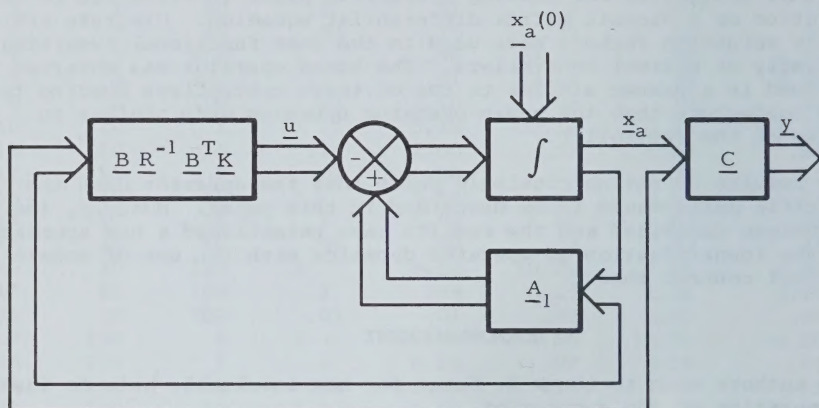


FIGURE 1. BLOCK DIAGRAM OF OPTIMAL REGULATOR

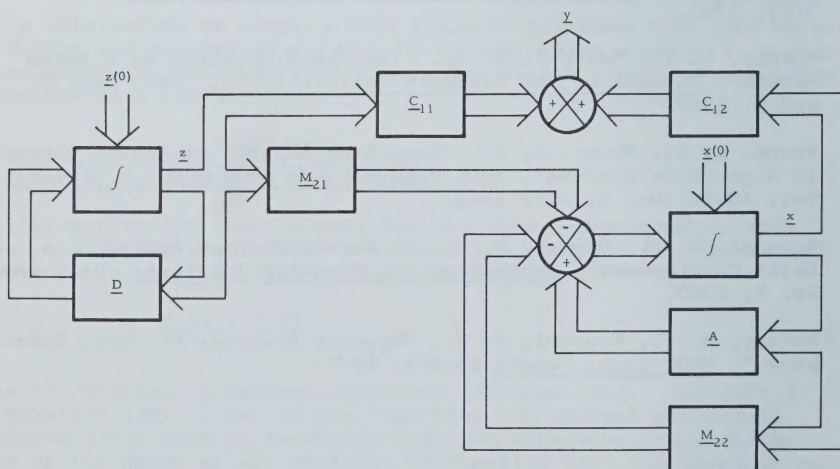


FIGURE 2. EXPANDED VECTOR REPRESENTATION OF OPTIMAL REGULATOR

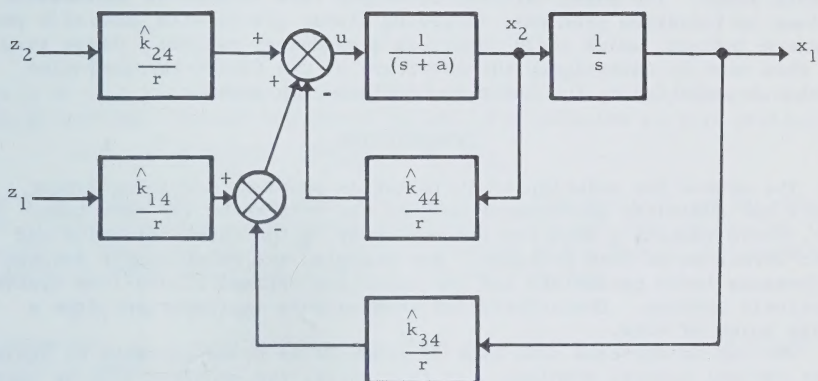


FIGURE 3. BLOCK DIAGRAM OF OPTIMAL TRACKER

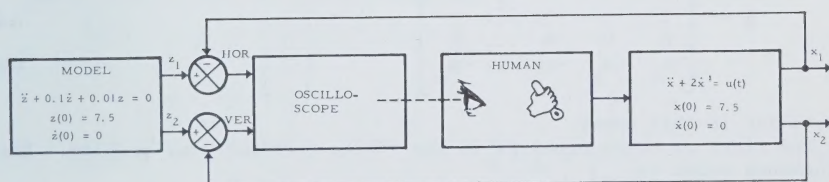


FIGURE 4. CONTROL TASK WITH HUMAN OPERATOR

SYSTEM EQUIVALENT OPTIMAL CONTROL PROBLEMS

W. J. RUGH and G. J. MURPHY
Department of Electrical Engineering
The Technological Institute
Northwestern University
Evanston, Illinois

ABSTRACT

Some new equivalence properties of optimal control problems are given in this paper. The phase-variable canonical form is used to delineate various optimization problems, involving linear plants with quadratic performance indices, which yield identical closed-loop systems. These results are then used to investigate the structure of the finite-terminal-time problem in relation to the infinite-terminal-time problem.

INTRODUCTION

The method for solution of optimization problems involving linear plants and quadratic performance indices has been known for some time. However, there remains a need for further study of the characteristics and basic structure of such problems. For example, the relationship between the performance index parameters and the resulting optimal closed-loop system is relatively unknown. Characteristics such as this are important from a design point of view.

One way to approach this type of study is by using concepts of equivalent optimal control problems. In this paper, two problems will be considered equivalent if they yield the same optimal closed-loop system. Two different relations are presented for both the finite-terminal-time and infinite-terminal-time problems where the plant is in phase-variable form. The results are then used to study the structure of the finite-terminal-time problem in relation to the infinite-terminal-time problem.

BACKGROUND

Two well known optimization problems relating to the control of a single input, time-invariant, controllable plant of the form

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b}u(t), \quad \underline{x}(0) = \underline{x}_0, \quad (1)$$

where

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_1 & -a_2 & -a_3 & \dots & -a_n \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \quad (2)$$

are studied in this paper.

The first of these problems is the finite-terminal-time problem. The performance index is

$$I = \frac{1}{2} \underline{x}'(T) \underline{F} \underline{x}(T) + \frac{1}{2} \int_0^T \{ \underline{x}'(t) \underline{Q} \underline{x}(t) + u^2(t) \} dt, \quad (3)$$

where $T < \infty$ is fixed and $\underline{F}$ and $\underline{Q}$ are symmetric, positive-semidefinite, $n \times n$ matrices. Kalman has shown [1] that the solution to this problem exists and is unique. The optimal control and the corresponding value of the performance index are given by

$$u^*(t) = -\underline{b}'\underline{K}(t) \underline{x}(t), \quad V^*(\underline{x}_0) = \frac{1}{2}\underline{x}_0'\underline{K}(0) \underline{x}_0, \quad (4)$$

respectively, where $\underline{K}(t)$, an $n \times n$ symmetric matrix, is the unique solution of the matrix Riccati equation

$$\dot{\underline{K}}(t) + \underline{K}(t) \underline{A} + \underline{A}' \underline{K}(t) - \underline{K}(t) \underline{b} \underline{b}' \underline{K}(t) + \underline{Q} = 0 \quad (5)$$

with

$$\underline{K}(T) = \underline{F}. \quad (6)$$

This problem is referred to in the sequel as an $(\underline{A}, \underline{Q}, \underline{F})$ problem.

The second problem to be studied is the infinite-terminal-time problem. The performance index is

$$I = \frac{1}{2} \int_0^\infty \{ \underline{x}'(t) \underline{Q} \underline{x}(t) + u^2(t) \} dt, \quad (7)$$

where $\underline{Q}$ is an $n \times n$ symmetric, positive-semidefinite matrix. This is termed an $(\underline{A}, \underline{Q})$ problem. Kalman has shown [2] that the solution to this problem is

$$u^*(t) = -\underline{b}'\underline{K} \underline{x}(t), \quad V^*(\underline{x}_0) = \frac{1}{2}\underline{x}_0' \underline{K} \underline{x}_0, \quad (8)$$

where $\underline{K}$ is a constant, symmetric, $n \times n$ matrix. The matrix $\underline{K}$ can be obtained by solving (5) with the boundary condition $\underline{F} = \underline{Q}$ and then evaluating the solution in the limit as $T \rightarrow \infty$. Also, it can be shown that $\underline{K}$ is a positive semidefinite solution of the algebraic Riccati equation

$$\underline{K} \underline{A} + \underline{A}' \underline{K} - \underline{K} \underline{b} \underline{b}' \underline{K} + \underline{Q} = 0. \quad (9)$$

Unfortunately, this equation does not have a unique positive-semidefinite solution in general.

This difficulty can be overcome in the following manner. If one writes $\underline{Q} = \underline{H}' \underline{H}$, where $\underline{H}$ is a $p \times n$ matrix and $p = \text{rank } \underline{Q}$, and makes the assumption that

$$\text{rank } [\underline{H}', \underline{A}' \underline{H}', \dots, (\underline{A}')^{n-1} \underline{H}'] = n, \quad (10)$$

it can be shown that $\underline{K}$ is positive-definite and is the unique positive-definite solution of (9). Further, with the assumption (10) it can be shown that the optimal closed-loop system characterized by

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b} u^*(t) = (\underline{A} - \underline{b} \underline{b}' \underline{K}) \underline{x}(t), \quad (11)$$

is asymptotically stable. That is, all the eigenvalues of $\underline{A} - \underline{b} \underline{b}' \underline{K}$ have negative real parts.

The assumption (10) will be mentioned when it is used in the sequel.

THE INFINITE-TERMINAL-TIME CASE

Let a second plant be characterized by

$$\dot{\hat{\underline{x}}}(t) = \hat{\underline{A}} \hat{\underline{x}}(t) + \hat{\underline{b}} \hat{u}(t), \quad \hat{\underline{x}}(0) = \hat{\underline{x}}_0, \quad (12)$$

where

$$\hat{\underline{A}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -\hat{a}_1 & -\hat{a}_2 & -\hat{a}_3 & \dots & -\hat{a}_n \end{bmatrix}, \quad \hat{\underline{b}} = \underline{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad (13)$$

and let the performance index associated with this plant be

$$\hat{I} = \frac{1}{2} \int_0^\infty \{ \hat{\underline{x}}'(t) \hat{\underline{Q}} \hat{\underline{x}}(t) + \hat{u}^2(t) \} dt, \quad (14)$$

where $\hat{Q}$ is a symmetric $n \times n$ matrix. (It is not assumed that $\hat{Q}$ is positive semidefinite.)

By similarity with the earlier $(\underline{A}, Q)$ problem, it is seen that the formal solution to this $(\hat{A}, \hat{Q})$ problem is

$$\hat{u}^*(t) = -\underline{b}' \hat{K} \hat{x}(t), \quad \hat{V}^*(\hat{x}_0) = \frac{1}{2} \hat{x}_0' \hat{K} \hat{x}_0, \quad (15)$$

where $\hat{K}$ is a symmetric $n \times n$ matrix that satisfies the equation

$$\hat{K} \hat{A} + \hat{A}' \hat{K} - \hat{K} \underline{b} \underline{b}' \hat{K} + \hat{Q} = \underline{0}. \quad (16)$$

The problem posed here is that of finding a relationship between $\underline{A}$, Q , $\hat{A}$, and $\hat{Q}$ such that if this relationship is satisfied, then the $(\underline{A}, Q)$ and $(\hat{A}, \hat{Q})$ problems yield identical optimal closed-loop systems. That is, then $\underline{A} - \underline{b} \underline{b}' \underline{K} = \hat{A} - \underline{b} \underline{b}' \hat{K}$.

To this end, let

$$\underline{D} = \begin{bmatrix} 0 & \dots & 0 & -a_1 + \hat{a}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & -a_{n-1} + \hat{a}_{n-1} \\ -a_1 + \hat{a}_1 & \dots & -a_{n-1} + \hat{a}_{n-1} & -a_n + \hat{a}_n \end{bmatrix} \quad (17)$$

and let

$$\underline{K} = \underline{K}_1 + \underline{K}_2 \quad (18)$$

where

$$\underline{K}_1 = \begin{bmatrix} k_{11} & \dots & k_{1,n-1} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ k_{1,n-1} & \dots & k_{n-1,n-1} & 0 \\ 0 & \dots & 0 & 0 \end{bmatrix} \quad \underline{K}_2 = \begin{bmatrix} 0 & \dots & 0 & k_{1n} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & k_{n-1,n} \\ k_{1n} & \dots & k_{n-1,n} & k_{nn} \end{bmatrix} \quad (19)$$

Also, let $\hat{K}$ be similarly defined as

$$\hat{K} = \hat{K}_1 + \hat{K}_2. \quad (20)$$

Now suppose that the following relationship holds:

$$\hat{Q} = Q + \underline{D} \hat{A} + \hat{A}' \underline{D} + \underline{D} \underline{b} \underline{b}' \underline{D}. \quad (21)$$

(Note that $\hat{Q}$ need not be positive-semidefinite.)

Theorem 1 The symmetric matrix $\underline{K} = \underline{K}_1 + \underline{K}_2$ is a solution of (9) if and only if the symmetric matrix $\hat{K} = \hat{K}_1 + \hat{K}_2$ with

$$\hat{K}_1 = \underline{K}_1, \quad \hat{K}_2 = \underline{K}_2 - \underline{D}, \quad (22)$$

is a solution of (16) with $\hat{Q}$ given by (21).

Proof See Appendix I.

This theorem gives a one-to-one correspondence between the solutions of the two Riccati equations. Of course, only one of the corresponding pairs $\underline{K}, \hat{K}$ is of interest: the pair in which $\underline{K}$ is the solution of the $(\underline{A}, Q)$ problem. It can be shown that (22) implies $\underline{A} - \underline{b} \underline{b}' \underline{K} = \hat{A} - \underline{b} \underline{b}' \hat{K}$ for each corresponding pair.

We now want to show that, given an $(\underline{A}, Q)$ problem, the matrix $\underline{Q}$ can be

replaced by a well defined diagonal matrix $\underline{Q}_d$ such that the two problems $(\underline{A}, \underline{Q})$ and $(\underline{A}, \underline{Q}_d)$ have the same optimal control law. Since the plants are the same, the two problems will thus yield identical optimal systems.

If $\underline{Q}$ is given as

$$\underline{Q} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{12} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{1n} & q_{2n} & \cdots & q_{nn} \end{bmatrix}, \quad (23)$$

the $(\underline{A}, \underline{Q}_d)$ problem is formed by setting

$$\underline{Q}_d = \begin{bmatrix} q_{11} & 0 & 0 & \cdots & 0 \\ 0 & q_{22} & -2q_{13} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & q_{nn} \end{bmatrix}. \quad (24)$$

More precisely, $\underline{Q}_d$ is a diagonal matrix, and the i^{th} diagonal element $q_{d_{ii}}$ is given by

$$q_{d_{ii}} = q_{ii} + 2 \sum_{j=1}^{k_i} (-1)^j q_{i-j, i+j}$$

(25)

where

$$k_i = \max \{j | 1 \leq i-j, i+j \leq n\}.$$

To solve this $(\underline{A}, \underline{Q}_d)$ problem formally, the Riccati equation

$$\underline{K}_d \underline{A} + \underline{A}' \underline{K}_d - \underline{K}_d \underline{b} \underline{b}' \underline{K}_d + \underline{Q}_d = 0 \quad (26)$$

must be solved for an $n \times n$ symmetric matrix $\underline{K}_d$.

Let $\underline{E}$ be an $n \times n$ symmetric matrix defined by

$$\underline{E} = \begin{bmatrix} q_{12} & q_{13} & \cdots & q_{1n} & 0 \\ q_{13} & q_{23} - q_{14} & \cdots & q_{2n} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ q_{1n} & q_{2n} & \cdots & q_{n-1, n} & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{12} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{1n} & e_{2n} & \cdots & e_{nn} \end{bmatrix}. \quad (27)$$

That is

$$e_{in} = 0, \quad i = 1, \dots, n$$

$$\text{where} \quad e_{ij} = \sum_{k=0}^{r_{ij}} (-1)^k q_{i-k, j+1+k}, \quad \begin{matrix} i \leq j \\ i = 1, \dots, n-1 \\ j = 1, \dots, n-1 \end{matrix} \quad (28)$$

$$r_{ij} = \max \{k | 1 \leq i-k, j+1+k \leq n\}.$$

In terms of this matrix, the solution can be expressed as in the following theorem.

Theorem 2 The symmetric matrix $\underline{K} = \underline{K}_1 + \underline{K}_2$ is a solution of (9) if and only if the symmetric matrix $\underline{K}_d = \underline{K}_{d1} + \underline{K}_{d2}$ defined by

$$\underline{K}_{d1} = \underline{K}_1 + \underline{E}, \quad \underline{K}_{d2} = \underline{K}_2 \quad (29)$$

is a solution of (26).

Proof See Appendix II.

This theorem gives a one-to-one correspondence between the solutions of the Riccati equations for the $(\underline{A}, \underline{Q})$ and $(\underline{A}, \underline{Q}_d)$ problems. Given an $(\underline{A}, \underline{Q})$ problem, $\underline{Q}_d$ can be written by inspection using (24). Then (26) can be solved for $\underline{K}_d$, and the corresponding $\underline{K}$ can be found from (29). Of course, the correct corresponding pair must be determined.

It should be noted that Wonham and Johnson pointed out [3] in 1963 that $\underline{Q}$ can be assumed diagonal for a class of problems in phase-variable form. However, they also stated that the elements of that diagonal matrix must be nonnegative for positive-semidefinite $\underline{Q}$. That this is not true was stated by Bass and Webber in 1965 [4] and can be shown by use of (24). In these earlier papers, integration by parts was used to accomplish the diagonalization and no explicit forms were determined.

The following theorem gives a useful property of $(\underline{A}, \underline{Q}_d)$ problems.

Theorem 3 The optimal control for an $(\underline{A}, \underline{Q}_d)$ problem is $u^*(t) = 0$ if and only if $\underline{Q}_d = \underline{0}$.

Proof See Appendix III.

These results are now applied to obtain a set of algebraic equations that determine the optimal closed-loop system for a given $(\underline{A}, \underline{Q})$ problem. To avoid difficulties in isolating the correct solution from several candidates, the assumption (10) is made for the given problem. This insures that the problem defines a unique asymptotically stable optimal system.

The general approach is to make use of the above results to convert the given problem into a problem that can be solved by inspection. Thus it can be seen that the equations which yield the optimal system arise naturally in this conversion.

For the given $(\underline{A}, \underline{Q})$ problem, a general $(\hat{\underline{A}}, \hat{\underline{Q}})$ problem is established, where $\hat{\underline{A}}$ is given in (13) and $\hat{\underline{Q}}$ in (21). The $(\hat{\underline{A}}, \hat{\underline{Q}}_d)$ problem is then established by use of (24). The diagonal matrix $\hat{\underline{Q}}_d$ has n nonzero terms involving the n variables $\hat{a}_i$, $i = 1, \dots, n$. If $\hat{\underline{Q}}_d$ is equated to $\underline{0}$ and the $\hat{a}_i$, $i = 1, \dots, n$, are chosen so as to satisfy the resulting n equations, then the optimal control for this $(\hat{\underline{A}}, \hat{\underline{Q}}_d = \underline{0})$ problem is $\hat{u}^*(t) = 0$, by Theorem 3. Thus, since the three problems lead to identical optimal systems,

$$\hat{\underline{A}} - \underline{b} \underline{b}' \hat{\underline{K}}_d = \hat{\underline{A}} = \underline{A} - \underline{b} \underline{b}' \underline{K}. \quad (30)$$

That is, $\hat{\underline{A}}$ described the optimal closed-loop system for the original $(\underline{A}, \underline{Q})$ problem. This is, of course, assuming that the solutions $\hat{a}_i$, $i = 1, \dots, n$, of $\hat{\underline{Q}}_d = \underline{0}$ are such that $\hat{\underline{A}}$ is asymptotically stable. That such a solution set exists and is unique follows from the one-to-one correspondences used in this approach.

THE FINITE-TERMINAL-TIME CASE

In this section some results similar to those given for the infinite-time problem are presented for the finite-time problem. First, an $(\hat{\underline{A}}, \hat{\underline{Q}}, \underline{F})$ problem similar in form to the $(\underline{A}, \underline{Q}, \underline{F})$ problem given previously is presented.

The plant is defined by (12) and (13), and the performance index is

$$\hat{I} = \frac{1}{2} \hat{\underline{x}}'(T) \hat{\underline{F}} \hat{\underline{x}}(T) + \frac{1}{2} \int_0^T \{ \hat{\underline{x}}'(t) \hat{\underline{Q}} \hat{\underline{x}}(t) + \hat{u}^2(t) \} dt. \quad (31)$$

It is assumed that $\hat{\underline{F}}$ and $\hat{\underline{Q}}$ are symmetric. The formal solution to this problem is

$$\hat{u}^*(t) = -\underline{b}' \hat{\underline{K}}(t) \hat{\underline{x}}(t), \quad \hat{V}^*(\underline{x}_0) = \frac{1}{2} \hat{\underline{x}}_0' \hat{\underline{K}}(0) \hat{\underline{x}}_0, \quad (32)$$

where $\hat{K}(t)$ is a symmetric $n \times n$ matrix and is a solution of

$$\dot{\hat{K}}(t) + \hat{K}(t) \hat{A} + \hat{A}' \hat{K}(t) - \hat{K}(t) \hat{b} \hat{b}' \hat{K}(t) + \hat{Q} = 0 \quad (33)$$

with

$$\hat{K}(T) = \hat{F}. \quad (34)$$

A particular relation involving $\hat{A}$, $\hat{Q}$, and $\hat{F}$ as well as the $(\hat{A}, \hat{Q}, \hat{F})$ problem such that the two problems yield identical optimal systems, is now specified by defining $\hat{D}$, $\hat{K}(t) = \hat{K}_1(t) + \hat{K}_2(t)$, and $\hat{K}(t) = \hat{K}_1(t) + \hat{K}_2(t)$ as in (17), (18), (19), and (20), and letting

$$\hat{Q} = Q + \bar{D} \hat{A} + \hat{A}' \bar{D} + \bar{D} \hat{b} \hat{b}' \bar{D} \quad (35)$$

and

$$\hat{F} = F - \bar{D}. \quad (36)$$

(Again, $\hat{Q}$ and $\hat{F}$ need not be positive-semidefinite.)

Theorem 4 The symmetric matrix $\hat{K}(t) = \hat{K}_1(t) + \hat{K}_2(t)$ is a solution of (5) with the boundary condition (6) if and only if the symmetric matrix $\hat{K}(t) = \hat{K}_1(t) + \hat{K}_2(t)$ defined by

$$\hat{K}_1(t) = K_1(t), \quad \hat{K}_2(t) = K_2(t) - \bar{D} \quad (37)$$

is a solution of (33) with the boundary condition (34), where $\hat{Q}$ and $\hat{F}$ are given by (35) and (36).

Proof The proof of this theorem is similar to that of Theorem 1 and is thus omitted.

Since the solution of (5) and (6) is unique and bounded on $[0, T]$, the one-to-one correspondence provided by this theorem implies that the solution to (33) and (34) is unique and, by (37), bounded on $[0, T]$. Thus the $(\hat{A}, \hat{Q}, \hat{F})$ problem has no conjugate points and represents a meaningful minimization problem [5].

It will now be noted that given an $(\hat{A}, \hat{Q}, \hat{F})$ problem, $\hat{Q}$ may be replaced by the diagonal matrix $\hat{Q}_d$ given in (24), and $\hat{F}$ by $\hat{F}_d = \hat{F} + \hat{E}$, where $\hat{E}$ is given in (27), such that the $(\hat{A}, \hat{Q}_d, \hat{F}_d)$ problem has the same optimal control law as the $(\hat{A}, \hat{Q}, \hat{F})$ problem. The Riccati equation for this new $(\hat{A}, \hat{Q}_d, \hat{F}_d)$ problem is

$$\dot{\hat{K}}_d(t) + \hat{K}_d(t) \hat{A} + \hat{A}' \hat{K}_d(t) - \hat{K}_d(t) \hat{b} \hat{b}' \hat{K}_d(t) + \hat{Q}_d = 0 \quad (38)$$

with

$$\hat{K}_d(T) = \hat{F}_d. \quad (39)$$

Theorem 5 The symmetric matrix $\hat{K}(t) = \hat{K}_1(t) + \hat{K}_2(t)$ is a solution of (5) and (6) if and only if the symmetric matrix $\hat{K}_d(t) = \hat{K}_{d1}(t) + \hat{K}_{d2}(t)$ defined by

$$\hat{K}_{d1}(t) = \hat{K}_1(t) + \hat{E}, \quad \hat{K}_{d2}(t) = \hat{K}_2(t) \quad (40)$$

is a solution of (38) and (39), where $\hat{Q}_d$ is given by (24) and $\hat{F}_d = \hat{F} + \hat{E}$, with $\hat{E}$ given in (27).

Proof The proof of this theorem is similar to that of Theorem 2 and is thus omitted.

Because of the one-to-one correspondence provided by this theorem, the equations (38) and (39) have a unique, bounded solution. Hence, the $(\hat{A}, \hat{Q}_d, \hat{F}_d)$ problem is a meaningful minimization problem.

The results of the previous sections are now used to investigate the structure of the finite-terminal-time problem. Suppose an $(\underline{A}, \underline{Q}, \underline{F})$ problem is given. For convenience, the assumption (10) is made for this problem.

Following the procedure used in the infinite-terminal-time case, a general $(\hat{\underline{A}}, \hat{\underline{Q}}, \hat{\underline{F}})$ problem is established by use of (35) and (36). Then the $(\hat{\underline{A}}, \hat{\underline{Q}}_d, \hat{\underline{F}}_d)$ problem can be established, where $\hat{\underline{Q}}_d$ is a diagonal matrix. Setting $\hat{\underline{Q}}_d$ equal to $\underline{Q}$ yields the same set of equations for the $\hat{a}_i$, $i = 1, \dots, n$, as an infinite-terminal-time $(\underline{A}, \underline{Q})$ problem. Because of the assumption (10), there is a unique solution for the $\hat{a}_i$, $i = 1, \dots, n$, for which $\hat{\underline{A}}$ is asymptotically stable. The given problem has now been reduced to the minimum control energy problem:

$$\dot{\hat{\underline{x}}}(t) = \hat{\underline{A}} \hat{\underline{x}}(t) + \hat{\underline{b}} \hat{u}(t) \quad (41)$$

$$\hat{\underline{I}} = \frac{1}{2} \hat{\underline{x}}'(T) \hat{\underline{F}}_d \hat{\underline{x}}(T) + \frac{1}{2} \int_0^T \hat{u}^2(t) dt, \quad (42)$$

where $\hat{\underline{A}}$ is the optimal system determined by the infinite-terminal-time problem

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b} u(t) \quad (43)$$

$$I = \frac{1}{2} \int_0^\infty \{ \underline{x}'(t) \underline{Q} \underline{x}(t) + u^2(t) \} dt. \quad (44)$$

The optimal system for the original $(\underline{A}, \underline{Q}, \underline{F})$ problem is identical to the optimal system resulting from the problem in (41) and (42).

Thus the finite-terminal-time problem has been decomposed into an infinite-terminal-time problem and a finite-terminal-time minimum control energy problem. Each optimal feedback gain for the $(\underline{A}, \underline{Q}, \underline{F})$ problem is given as the sum of a constant gain from the infinite-time problem and a time-varying gain from the minimum control energy problem.

This decomposition explicitly displays the relationship between the finite-time and infinite-time optimization problems. In fact, the finite time problem may be considered as an infinite-time problem with a finite time minimum control energy "perturbation" problem. (Note that in this minimum control energy problem, $\hat{\underline{F}}_d$ need not be positive-semidefinite.)

In a recent paper, [6] Porter has shown that under certain assumptions a relatively simple explicit solution can be obtained for finite-time minimum control energy problems. For systems of low order, where it may be desirable to see the explicit structure of the optimal control law, the approach described herein may be quite useful in conjunction with Porter's results.

Example

Consider the $(\underline{A}, \underline{Q}, \underline{F})$ problem

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -2.45 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad (45)$$

$$I = \frac{1}{2} \underline{x}'(T) \begin{bmatrix} 7 & 2 \\ 2 & 1.1 \end{bmatrix} \underline{x}(T) + \frac{1}{2} \int_0^T \{ \underline{x}'(t) \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \underline{x}(t) + u^2(t) \} dt. \quad (46)$$

It is easily verified that the assumption (10) holds for this problem. The system equivalent $(\hat{\underline{A}}, \hat{\underline{Q}}, \hat{\underline{F}})$ problem is, from (35) and (36),

$$\dot{\hat{\underline{x}}}(t) = \begin{bmatrix} 0 & 1 \\ -\hat{a}_1 & -\hat{a}_2 \end{bmatrix} \hat{\underline{x}}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \hat{u}(t), \quad (47)$$

$$\hat{I} = \frac{1}{2} \hat{\underline{x}}'(T) \begin{bmatrix} 7 & 2-\hat{a}_1 \\ 2-\hat{a}_1 & 3.55-\hat{a}_2 \end{bmatrix} \hat{\underline{x}}(T) + \frac{1}{2} \int_0^T \{ \hat{\underline{x}}'(t) \begin{bmatrix} 4-\hat{a}_1^2 & -\hat{a}_1 \hat{a}_2 \\ -\hat{a}_1 \hat{a}_2 & 8-\hat{a}_2^2+2\hat{a}_1 \end{bmatrix} \hat{\underline{x}}(t) + \hat{u}^2(t) \} dt. \quad (48)$$

Using (24) and (27), the system equivalent to the $(\hat{A}, \hat{Q}_d, \hat{F}_d)$ problem is specified by the plant (47) and the performance index

$$\hat{I}_d = \frac{1}{2} \hat{\underline{x}}'(T) \begin{bmatrix} 7-\hat{a}_1 \hat{a}_2 & 2-\hat{a}_1 \\ 2-\hat{a}_1 & 3.55-\hat{a}_2 \end{bmatrix} \hat{\underline{x}}(T) + \frac{1}{2} \int_0^T \{ \hat{\underline{x}}'(t) \begin{bmatrix} 4-\hat{a}_1^2 & 0 \\ 0 & 8-\hat{a}_2^2+2\hat{a}_1 \end{bmatrix} \hat{\underline{x}}(t) + \hat{u}^2(t) \} dt. \quad (49)$$

Setting $\hat{Q}_d$ equal to $\underline{0}$ and solving yields the asymptotically stable matrix

$$\hat{\underline{A}} = \begin{bmatrix} 0 & 1 \\ -\hat{a}_1 & -\hat{a}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3.46 \end{bmatrix}.$$

Note that, according to the theory of the infinite-terminal-time case, $\hat{\underline{A}}$ is the optimal system for the infinite-time $(\hat{A}, \hat{Q})$ problem. That is, the problem specified by the plant (45) and the performance index

$$I = \frac{1}{2} \int_0^\infty \{ \underline{x}'(t) \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \underline{x}(t) + u^2(t) \} dt.$$

Now, the minimum control energy problem which is system equivalent to the original $(\underline{A}, \underline{Q}, \underline{F})$ problem is

$$\dot{\hat{\underline{x}}}(t) = \begin{bmatrix} 0 & 1 \\ -2 & -3.46 \end{bmatrix} \hat{\underline{x}}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \hat{u}(t) \quad (50)$$

$$\hat{I} = \frac{1}{2} \hat{\underline{x}}'(T) \begin{bmatrix} .08 & 0 \\ 0 & .09 \end{bmatrix} \hat{\underline{x}}(T) + \frac{1}{2} \int_0^T \hat{u}^2(t) dt. \quad (51)$$

From the numerical values in the performance index (51), it seems clear that, since the plant (50) is asymptotically stable, the time-varying feedback gains would be small in magnitude (depending somewhat on T). This conclusion can be substantiated by applying the result of [6] to solve this problem analytically. Thus, in this case, the finite-time solution is very little different from the infinite-time solution.

REFERENCES

1. Kalman, R. E.: "Contributions to the theory of optimal control", Bol. Soc. Mat. Mex., Vol. 5, 1960, pp. 102-119
2. Kalman, R. E.: "When is a linear control system optimal?", ASME J. Basic Engr., Vol. 86, 1964, pp. 51-60
3. Wonham, W. M., and Johnson, C. D.: "Optimal bang-bang control with a quadratic performance index", ASME J. Basic Engrg., Vol. 86, 1964, pp. 107-115
4. Bass, R. W. and Webber, R. F.: "On synthesis of optimal bang-bang feed-back control systems with quadratic performance index", JACC Preprints, 1965, pp. 213-219
5. Breakwell, J. V. and Ho, Y. C.: "On the conjugate point condition for the control problem", Internat. J. Engr. Sci., Vol. 2, 1965, pp. 565-580
6. Porter, W. A.: "On the matrix Riccati equation", IEEE Trans. on A. C., Vol. AC-12, 1967, pp. 746-749

APPENDICES

Appendix I

Proof of Theorem 1 : Suppose $\underline{K} = \underline{K}_1 + \underline{K}_2$ is a solution of (9). Then (9) can be written as

$$\underline{K}_1 \underline{A} + \underline{K}_2 \underline{A} + \underline{A}' \underline{K}_1 + \underline{A}' \underline{K}_2 - \underline{K}_2 \underline{b} \underline{b}' \underline{K}_2 + \underline{Q} = \underline{0}. \quad (52)$$

From (17), it is clear that $\underline{A} = \hat{\underline{A}} + \underline{b} \underline{b}' \underline{\bar{D}}$ and hence, since $\underline{K}_1 \underline{b} \underline{b}' \underline{\bar{D}} = \underline{\bar{D}} \underline{b} \underline{b}' \underline{K}_1 = \underline{0}$,

$$\underline{K}_1 \hat{\underline{A}} + \hat{\underline{A}}' \underline{K}_1 + \underline{K}_2 \hat{\underline{A}} + \underline{K}_2 \underline{b} \underline{b}' \underline{\bar{D}} + \hat{\underline{A}}' \underline{K}_2 + \underline{\bar{D}} \underline{b} \underline{b}' \underline{K}_2 - \underline{K}_2 \underline{b} \underline{b}' \underline{K}_2 + \underline{Q} = \underline{0}. \quad (53)$$

Now, using (21) to substitute for $\underline{Q}$ and collecting linear and quadratic terms gives

$$\underline{K}_1 \hat{\underline{A}} + (\underline{K}_2 - \underline{\bar{D}}) \hat{\underline{A}} + \hat{\underline{A}}' \underline{K}_1 + \hat{\underline{A}}' (\underline{K}_2 - \underline{\bar{D}}) - (\underline{K}_2 - \underline{\bar{D}}) \underline{b} \underline{b}' (\underline{K}_2 - \underline{\bar{D}}) + \hat{\underline{Q}} = \underline{0}. \quad (54)$$

From (22), it is clear that

$$\hat{\underline{K}} \hat{\underline{A}} + \hat{\underline{A}}' \hat{\underline{K}} - \hat{\underline{K}} \underline{b} \underline{b}' \hat{\underline{K}} + \hat{\underline{Q}} = \underline{0}. \quad (55)$$

To complete the proof, it suffices to note that the above procedure can be reversed.

Appendix II

Proof of Theorem 2 : If $\underline{K} = \underline{K}_1 + \underline{K}_2$ is a solution of (9), then (52) holds. Substituting (29) into (52) gives

$$\underline{K}_{d1} \underline{A} - \underline{E} \underline{A} + \underline{A}' \underline{K}_{d1} - \underline{A}' \underline{E} + \underline{K}_{d2} \underline{A} + \underline{A}' \underline{K}_{d2} - \underline{K}_{d2} \underline{b} \underline{b}' \underline{K}_{d2} + \underline{Q} = \underline{0}, \quad (56)$$

or

$$\underline{K}_d \underline{A} + \underline{A}' \underline{K}_d - \underline{K}_d \underline{b} \underline{b}' \underline{K}_d + (\underline{Q} - \underline{E} \underline{A} - \underline{A}' \underline{E}) = \underline{0}. \quad (57)$$

It remains to be shown that

$$\underline{Q}_d = \underline{Q} - \underline{E} \underline{A} - \underline{A}' \underline{E}. \quad (58)$$

Although this identity can be confirmed by developing a general form for each element using (23), (24), (25), and (28), this procedure is quite cumbersome and will be omitted here. We simply note that (58) can easily be confirmed by writing out the equation in general form.

To complete the proof, the above process is reversed.

Appendix III

Proof of Theorem 3 : Suppose that an $(\underline{A}, \underline{Q}_d = \underline{0})$ problem is given. That is, the performance index is

$$I = \frac{1}{2} \int_0^{\infty} u^2(t) dt. \quad (59)$$

Then, clearly, I is minimized by $u^*(t) = 0$.

Conversely, suppose it is known that $u^*(t) = 0$ for a certain $(\underline{A}, \underline{Q}_d)$ problem. Then it is easily seen that $\underline{K}_{d2} = \underline{0}$, and the Riccati equation for this problem can be written as

$$\underline{K}_{d1} \underline{A} + \underline{A}' \underline{K}_{d1} + \underline{Q}_d = \underline{0}. \quad (60)$$

Again, it is left to the reader to show that (60) implies $\underline{Q}_d = \underline{0}$ (and, indeed, $\underline{K}_{d1} = \underline{0}$) by writing this equation out in general form.

ACKNOWLEDGMENT

The work described in this paper was supported in part by the U.S. Army, Navy, and Air Force in a Joint Service Electronics Program under Office of Naval Research Contract Number N00014-67-A-0356-0003, Modifications AB, and in part by the Advanced Research Projects Agency of the Department of Defense under the same contract, monitored by the Office of Naval Research.

N. K. SINHA and V. TEMPLE

McMaster University, Hamilton, Ontario, Canada.

Abstract: A new cost function is proposed which is more suitable for implementing adaptive control without complete plant identification than the conventional integral criteria. This cost function is a vector with as many dimensions as the number of variable parameters in the controller and is available at all instants of time.

Summary: Adaptive control can be implemented without complete identification of the dynamic characteristics of the plant to be controlled if the cost function is measured continually and the parameters of the controller are adjusted until the minimum of the cost function is reached. The main difficulty with this approach is that most of the cost functions proposed for adaptive systems are of the integral type, so that the time between adjustments is considerable, and may even lead to instability. The general integral criterion is of the form:

$$J = \int_{t_0}^{t_f} W(\underline{u}, \underline{x}, \underline{z}, t) dt \quad (1)$$

where W is a scalar non-negative definite function of the m -dimensional input vector $\underline{u}$, the n -dimensional actual and desired output vectors $\underline{x}$ and $\underline{z}$, and time t , with t_0 and t_f being the initial and final times respectively.

Changing the upper limit of integration in (1) to t provides the function $J(t)$, which indicates the cost at all instants of time, but this is of limited usefulness. A more suitable function is obtained by defining a vector $\underline{h}(t)$ in such a manner that the scalar product $\langle \underline{h}, \underline{h} \rangle$ varies monotonically, and is one-to-one with W . In most cases it will be possible to select $\underline{h}$ such that $\langle \underline{h}, \underline{h} \rangle$ can be equal to W . Let $\underline{h}^*(t)$ be that value of $\underline{h}$ which yields the optimal W , and define:

$$\underline{g}(t) = \underline{h}(t) - \underline{h}^*(t) \quad (2)$$

It follows that if $\langle \underline{h}, \underline{h} \rangle$ is minimized in each time interval Δt , a local minimum of J is obtained. But if $\underline{h}^*$ can be found, minimizing $\langle \underline{g}, \underline{g} \rangle$ assures a globally minimum cost function. Although $\underline{h}^*$ may not always be obtainable, a vector cost function can still be used to advantage in a learning and adaptive loop, the controllers driving the systems to a locally minimum cost function, by simply putting $\underline{h}^*(t) = 0$.

Consider the Taylor's series expansion:

$$g(t + \Delta t) = g(t) + \frac{dg(t)}{dt} \Delta t + \dots + d^n \frac{g(t)}{dt^n} \frac{(\Delta t)^n}{n!} + \dots \quad (3)$$

Define

$$g^{(n)}(t) = \frac{d^{n-1}}{dt^{n-1}} g(t) \cdot \frac{(\Delta t)^{n-1}}{(n-1)!} \quad (4)$$

and the generalized cost function:

$$\underline{G}(t) = \begin{bmatrix} g^{(1)} \\ g^{(2)} \\ \vdots \\ g^{(N)} \end{bmatrix} \quad (5)$$

with N times as many dimensions as in g . Minimizing $\underline{G}(t)$ requires a controller of at least as many adjustable parameters as the number of dimensions in $\underline{G}(t)$ in order to drive $g(t)$ and its N-1 derivatives to zero.

Let $\underline{c}$, $\underline{a}$ and $\underline{f}$ represent the vectors of the variable parameters of the forward loop controller, the plant, and the feedback loop controller, respectively. The following Taylor's series expansion of the vector cost function is now obtained:

$$\underline{G}(t + \Delta t) = \underline{G}(t) + s_t \Delta t + s_c \Delta \underline{c} + s_a \Delta \underline{a} + s_f \Delta \underline{f} \quad (6)$$

where

$$\begin{aligned} s_t &= \frac{\partial \underline{G}}{\partial t} & , & & s_c &= \frac{\partial \underline{G}}{\partial \underline{c}} \\ s_a &= \frac{\partial \underline{G}}{\partial \underline{a}} & , & & s_f &= \frac{\partial \underline{G}}{\partial \underline{f}} \end{aligned} \quad (7)$$

be defined as the sensitivity matrices of the vector cost function with respect to t , $\underline{c}$, $\underline{a}$ and $\underline{f}$, respectively.

Equation (7) will be called "the adaptor equation". An adaptation scheme based on (7) was shown to be effective in the adaptive control of plants with varying parameter by means of a number of experiments^{1,2}. This continuous minimization of $\underline{G}(t)$ may be viewed as analogous to the continuous maximization of the Hamiltonian in applying Pontryagin's Minimum Principle.

Moreover, taking the first partial derivative of $\underline{G}$ with respect to a variable parameter implies truncating the Taylor's series expansion of $g(t)$ after N derivatives. Hence, some difficulties experienced with the use of first-order sensitivity terms are overcome³.

References:

1. V. Temple, "Nth Order Self-Adaptive Systems", M.Eng. Thesis, McMaster University, October 1967.
2. V. Temple and N. K. Sinha, "A Vector Cost Function for Efficient Adaptation", (to be published).
3. N. K. Sinha and S. R. Atluri, "Sensitivity of Optimal Control Systems", Proceedings of the Fourth Allerton Conference on Circuit and System Theory, 1966, pp. 508 - 516.

JURAJ MEDANIC*
 Coordinated Science Laboratory
 University of Illinois
 Urbana, Illinois 61801

ABSTRACT

An algorithm based on elimination methods of nonlinear programming is presented for the solution of the minimax problem. The procedure reduces the admissible domain of minimizing parameters to an arbitrarily small neighborhood of the minimax value.

INTRODUCTION

Interest in differential games and control of uncertain systems has lead to a corresponding interest in the solution of the minimax problem. Recently necessary conditions for the minimax, which may serve for the development of iterative procedures have been obtained by Danskin,² Bram³ and Demjenov.⁴ However, at the present time, these procedures are still in their formative stage. A computational solution has recently been proposed by Salmon.⁷ In this paper the author proposes an elimination algorithm for the solution of the minimax problem. In order to present the salient features of the algorithm the following problem in finite dimension spaces is considered: Find (c^*, p^*) such that

$$J(c^*, p^*) = \min_{c \in C} \max_{p \in P} J(c, p), \quad (1)$$

where $J(c, p)$ is defined and continuously differentiable in a finite dimensional convex, closed and bounded domain $C \times P$. The minimax problem is solved under the assumption that $J(c, p)$ is convex in C for all $p \in P$.

Due to the ambiguous usage of the term minimax in some recent references it is necessary to state at the outset that here the term minimax problem is reserved for problems where the minimax need not at the same time be a saddle point. Under the convexity assumption made above the solution of the minimax problem is not (although in certain cases it may be) a saddle point, since concavity of $J(c, p)$ in P has not been assumed.¹

The contour gradient idea is used to construct a hyperplane dividing the admissible domain into two subdomains, one of which may be eliminated since the minimax solution must be in the other. Hence, a basic

* On leave from Mihailo Pupin Institute, Belgrade, Yugoslavia.

contribution here is in ascertaining that elimination type algorithms are suitable for the solution of the minimax problem. However, although the elimination idea has been proposed in minimization problems, where its validity is evident, it has not been developed into an algorithm. (An important reason, aside from specific computational requirements, is that numerous other procedures exist for finding the minimum of a convex function over a convex admissible domain. This, however, is not true in the minimax problem.) In the second part of this paper a systematic elimination algorithm is developed. Its basic computational requirements demand the solution of a set of linear equations and the maximization of $J(c, p)$ over P for given $c \in C$. This is considerably less than demanded by other proposals. Still, in section four, it is shown that the number of eliminations in which the maximization must be carried out can be reduced, depending on the initial guess $c_1 \in C$.

ADAPTATION OF THE CONTOUR GRADIENT METHOD

Consider the contour gradient idea⁵ in the minimization of a convex function $F(c)$, where c is restricted to a convex domain C . Select $c_n \in C$; expand $F(c)$ about c_n . Due to convexity,

$$F(c) \geq F(c_n) + \frac{\partial F}{\partial c_n} (c - c_n). \quad (2)$$

Whenever c belongs to the domain $\frac{\partial F}{\partial c_n} (c - c_n) > 0$, $F(c) > F(c_n)$. Hence the minimum of $F(c)$ in C cannot belong to the subdomain $\frac{\partial F}{\partial c_n} (c - c_n) > 0$ and this subdomain can be eliminated. A new point c_{n+1} belonging to the intersection of $\frac{\partial F}{\partial c_n} (c - c_n) \leq 0$ and C is selected and the procedure repeated.

The same idea can be employed in the solution of the more complicated minimax problem by recalling the convexity assumption made about $J(c, p)$. In implementing the idea it is necessary only to make a distinction between the roles of c and p . It is then possible to find a particular hyperplane through the selected point $c_n \in C$ which divides C into two domains, one of which may be eliminated since the minimax value c^* must be in the other. In the minimax problem, then, the algorithm will reduce the domain C to an arbitrarily small neighborhood of the minimax point c^* . The role of maximizing parameters is to characterize the correct hyperplane through c^* .

Consider the basic steps in the n -th elimination. Select $c_n \in C^n$, where C^n is either C (when $n=1$) or that part of C where, in view of previous eliminations, the minimax point c^* can still be located.

Maximize with respect to p :

$$J(c_n, p_n) = \max_{p \in P} J(c_n, p). \quad (3)$$

If $J(c_n, p_n)$ is not the minimax,

$$J(c_n, p_n) > \min_{c \in C} \max_{p \in P} J(c, p). \quad (4)$$

Expand $J(c, p_n)$ about c_n ; $J(c, p_n)$ is convex in C and hence

$$J(c, p_n) \geq J(c_n, p_n) + g_n^*(c - c_n) \quad (5)$$

where $g_n^* = \left. \frac{\partial J}{\partial c} \right|_{c=c_n, p=p_n}$; here "*" denotes transpose. Employing the contour gradient idea it follows that

$$J(c, p_n) > J(c_n, p_n) \quad (6)$$

in the subdomain where

$$g_n^*(c - c_n) > 0. \quad (7)$$

It remains to be shown that (6) and (4) imply that the minimax (c^*, p^*) cannot be contained in the halfspace (7). Indeed, for all c_k belonging to the halfspace (7) we have

$$J(c_k, p_k) = \max_{p \in P} J(c_k, p) \geq J(c_k, p_n) > J(c_n, p_n) > \min_{c \in C} \max_{p \in P} J(c, p) \quad (8)$$

Equivalently, the minimax point $c^* \in C$ can be contained only in the intersection of the half-space

$$g_n^*(c - c_n) \leq 0 \quad (9)$$

with the admissible domain C^n .

ELIMINATION ALGORITHM

To construct an elimination algorithm it is necessary to define a feasible method of selecting nominal points from the remaining part of the admissible domain and to determine a viable stopping criterion. The following algorithm is proposed in this paper.

At the beginning of the n -th elimination, let the reduced domain C^n be defined by the set of linear constraints

$$g_i^* c \leq g_i^* c_i, \quad i=1, \dots, m+n-1 \quad (10)$$

where for $i=1, \dots, m$, the inequalities correspond to constraints originally defining C while the rest have been added in the previous $n-1$ eliminations.

Let the point c_n be the nominal point obtained in the previous elimination. Maximize $J(c_n, p)$ with respect to p as in (3) and find the contour gradient in the sense of (9). This adds the constraint

$$g_n^* c \leq g_n^* c_n \quad (11)$$

to (10) and reduces C^n to C^{n+1} . To select a nominal point in C^{n+1} observe that all points of the form

$$c = c_n - \lambda g_n \quad (12)$$

are in C^{n+1} , for positive and sufficiently small λ (such that none of the constraints in (10) are violated). It is proposed that the nominal point be selected as the midpoint between c_n and the first intersection of the ray (12) with the constraint set (10). The value of λ corresponding to the point of intersection is

$$\lambda_n = \min_i \frac{g_i^*(c_n - c_i)}{g_i^* g_n} \quad , \quad i=1, \dots, m+n-1 \quad (13)$$

and $\lambda_i > 0$. Hence, the new nominal point is

$$c_{n+1} = c_n - \frac{1}{2} \lambda_n g_n. \quad (14)$$

The step size between c_n and c_{n+1} is selected as the basic stopping criterion. The procedure is interrupted when the step size is below a prespecified bound q , i.e., when

$$\lambda_n = \frac{q}{\|g_n\|} \quad (15)$$

An additional stopping criterion should be employed before the search is completely stopped. The quadratic criterion employed here is a measure of the volume of C^n ; it is therefore a measure of the precision with which the minimax has been located. If after N iterations (15) becomes satisfied, reduce the set of constraints (10), (11) to the set of constraints

$$Gc \leq h \quad (16)$$

where only those inequalities in (10) which form the boundary of C^N are included into (16) (i.e., the set of linear inequalities defining C^N is brought to its irreducible form). Hence,

$$G = \begin{bmatrix} g_1^* \\ \vdots \\ g_k^* \end{bmatrix}, \quad h = \begin{bmatrix} g_1^* c_1 \\ \vdots \\ g_k^* c_k \end{bmatrix}, \quad (17)$$

relabeling the inequalities if necessary. Consider now the criterion

$$S = (Gc-h)^*(Gc-h) \quad (18)$$

and find the minimizing c , given by

$$c_{N+1} = (G^*G)^{-1}G^*h \quad (19)$$

as well as the corresponding minimal value of S , equal to

$$S_{\min} = h^*[I-G(G^*G)^{-1}G^*]h. \quad (20)$$

The search for the minimax value c^* is completed if aside from (15)

$$S_{\min} \leq d \quad (21)$$

is also satisfied, where d is a prespecified bound. If (21) is not satisfied select c_{N+1} given by (20) for the new nominal point and continue the eliminations with (16) now defining the admissible domain of C .

EXAMPLE

The simple example presented for illustratory purposes is not trivial since it has all characteristics of the general minimax problem: The minimizing c^* is in the interior of C and the max function is non-differentiable at the minimum. The following quadratic function was considered:

$$J(c,p) = c_1^2 + (\frac{1}{2} - p_1)c_1 + c_2^2 + (\frac{2}{3} - p_2)c_2 + \frac{1}{2} p_1^2 + \frac{1}{4} p_2^2$$

with

$$0 \leq c_i \leq 1, \quad 0 \leq p_i \leq 1, \quad i=1,2.$$

Hence constraints defining C are of the form (10) with $i=2,b,c,d$, where

$$g_a = \begin{vmatrix} 1 \\ 0 \end{vmatrix}, \quad g_b = \begin{vmatrix} 0 \\ 1 \end{vmatrix}, \quad g_c = \begin{vmatrix} -1 \\ 0 \end{vmatrix}, \quad g_d = \begin{vmatrix} 0 \\ -1 \end{vmatrix}$$

$$c_a = \begin{vmatrix} 1 \\ 0 \end{vmatrix}, \quad c_b = \begin{vmatrix} 0 \\ 1 \end{vmatrix}, \quad c_c = \begin{vmatrix} 0 \\ 0 \end{vmatrix}, \quad c_d = \begin{vmatrix} 0 \\ 0 \end{vmatrix}.$$

Select the initial point $c_1 = (\frac{1}{2} \frac{1}{2})^*$. Maximization of $J(c_1,p)$ produces two maximizing values: $p_1^1 = (0 \ 0)^*$ and $p_1^2 = (0 \ 1)^*$. We make use of only p_1^1 . The contour gradient corresponding to (c_1, p_1^1) is of the form (11), where $c_1 = (\frac{1}{2} \frac{1}{2})^*$ and $g_1 = (\frac{3}{2} \frac{7}{6})^*$. To find c_2 we need λ_1 , which is equal to

$$\lambda_1 = \min_j \frac{g_1^*(c_1 - c_j)}{g_j^* g_1} = \frac{g_c^*(c_1 - c_c)}{g_c^* g_1} = \frac{1}{3}.$$

Hence, from (14), $c_2 = (\frac{1}{2} \frac{3}{10})^*$ and the procedure is now repeated.

Figure 1 illustrates the results of the first six eliminations. Dotted lines are rays from nominal points, full lines are contour gradient hyperplanes and eliminated portions in each elimination are shaded. With six complete eliminations the area of C has been reduced to 0.75% of the original size. The minimax is obtained at

$$c^* = \begin{pmatrix} \frac{3}{4} \\ \frac{1}{6} \end{pmatrix}$$

with either of the two values for p :

$$p^{1*} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad p^{2*} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

It should be noted that two maximizing elements $p_n^1, p_n^2 (n=1, \dots, 4)$ were obtained in the first four eliminations in this particular example. In the algorithm used here only one of the two points was employed. A larger portion of C^n is eliminated, per elimination, if in each elimination all maximizing elements $\{p_n^j\}$ are used since to each p_n^j corresponds a different hyperplane through c_n . However, the selection of nominal points from elimination to elimination becomes more complicated. If only one maximizing element is employed the algorithm is straightforward as illustrated above. One may wish, however, to select the best element from $\{p_n^j\}$. This cannot be achieved without the understanding of the particular problem being considered since the direction of hyperplanes with respect to the shape of the domain C^n is important. However, one may expect that with a larger step size between nominal points the chances for greater reduction are increased. Hence, for each p_n^j a corresponding λ_n^j is found from (14). From the set $\{p_n^j\}$ one then selects that element to which there corresponds the greatest step size between c_n and c_{n+1}^j .

MODIFICATION OF THE ALGORITHM

Maximization of $J(c_n, p)$ in P is a significant computational task in the elimination algorithm. While it cannot be avoided altogether, it is possible to reduce the number of eliminations in which a complete maximization must be performed. In the minimax problem this comes as a result of the possible existence of a family of usable hyperplanes through the nominal point c_n . This in turn depends on the data collected in prior eliminations, and ultimately on a good initial guess for c_1 . Let us consider this in detail. Suppose $J(c_n, p_n)$ is the smallest element in the set $\{J(c_k, p_k) | k=1, \dots, n\}$, obtained either through the use of the algorithm in part two or through the modification now discussed. That is, $J(c_n, p_n)$

is the smallest maximum obtained by the algorithm in the first n eliminations. Let c_{n+1} be the new nominal point, and let

$$J(c_{n+1}, p_n) > J(c_n, p_n). \quad (22)$$

The expansion of $J(c, p_n)$ about c_{n+1} leads to

$$J(c, p_n) \geq J(c_{n+1}, p_n) + \frac{\partial J}{\partial c}(c_{n+1}, p_n)(c - c_{n+1}), \quad (23)$$

and it follows from (23), (22) and

$$\max_{p \in P} J(c, p_n) \geq J(c, p_n) \quad (24)$$

that the minimax point c^* cannot be contained in the half-space

$$\frac{\partial J}{\partial c}(c_{n+1}, p_n)(c - c_{n+1}) > 0. \quad (25)$$

Hence, the constraint

$$\frac{\partial J}{\partial c}(c_{n+1}, p_n)(c - c_{n+1}) \leq 0, \quad (26)$$

instead of (9), may be used to eliminate a part of C^n . The rest of the algorithm remains unaffected. Likewise, if $J(c_{n+1}, p_n) < J(c_n, p_n)$, it is necessary to move towards the maximum in P only until some point p_k is reached such that $J(c_{n+1}, p_k) > J(c_n, p_n)$, where upon, by repeating the arguments above, the hyperplane

$$\frac{\partial J}{\partial c}(c_{n+1}, p_k)(c - c_{n+1}) \leq 0 \quad (27)$$

is used to eliminate a portion of C^n . It follows that the full maximization of $J(c_{n+1}, p_n)$ must be performed only when $J(c_{n+1}, p_{n+1}) = \max_{p \in P} J(c_{n+1}, p) < J(c_n, p_n)$. In that case there is only one usable hyperplane through c_{n+1} and it is given by (9) (with relabeling the index to $n+1$).

CONCLUSIONS

The elimination method for the solution of the minimax problem leads to an algorithm which finds the minimax point $c^* \in C$ by reducing the admissible domain C to an arbitrarily small neighborhood of c^* . The procedure is relatively simple and converges to the minimax with arbitrary selection of nominal points within the remaining part of C . An algorithm presented in the paper employs a specific method of selecting nominal points. The algorithm is not restricted to the problems where C is defined by linear constraints as in (10). However, nonlinear boundaries of C may,

for the purpose of implementing the algorithm, be represented by a suitable number of supporting hyperplanes. Since this representation is carried out only for the purpose of selecting nominal points it does not affect the final result. Moreover, due to the nature of the elimination algorithm, the nonlinear constraints are constantly being replaced by linear constraints, i.e. contour gradients. Extension of the algorithm to parameter optimization problems in dynamic systems where the vector p is then either a set of uncertain parameters or a disturbance vector is straightforward. A serious limitation of the elimination method is that $J(c,p)$ must be convex in C . This limitation is unfortunate since it limits the generality of the method and excludes direct application to the general minimax problem.

ACKNOWLEDGMENT

This work was supported in part by the Joint Services Electronics Program (U. S. Army, U. S. Navy, and U. S. Air Force) under Contract No. DAAB-07-67-C-0199, and a National Science Foundation Grant.

REFERENCES

1. Karlin, S., Mathematical Methods and Theory in Games, Programming and Economics, Vol. 1, Addison-Wesley, 1959, Reading, Mass.
2. Danskin, J. M., The Theory of Max-Min, Series in Econometrics and Operations Research, Vol. V, Springer-Verlag, New York, 1967.
3. Bram, J., "The Lagrange Multiplier Theorem for Min-Max with Several Constraints," SIAM Journal, Vol. 14, 1966, pp. 665-667.
4. Demjenov, V. F., Kibernetika, Vol. 2, No. 6, 1966.
5. Wilde, D. and C. Beightler, Foundations of Optimization, Prentice-Hall, 1967.
6. Newman, D. J., "Location of the Maximum on Unimodal Surfaces," J. Assoc. Comp. Mach., Vol. 12, No. 3, 1965.
7. Salmon, D., "Minimax Controller Design," IEEE Trans. on Automatic Control, Vol. AC-13, No. 3, July, 1968.

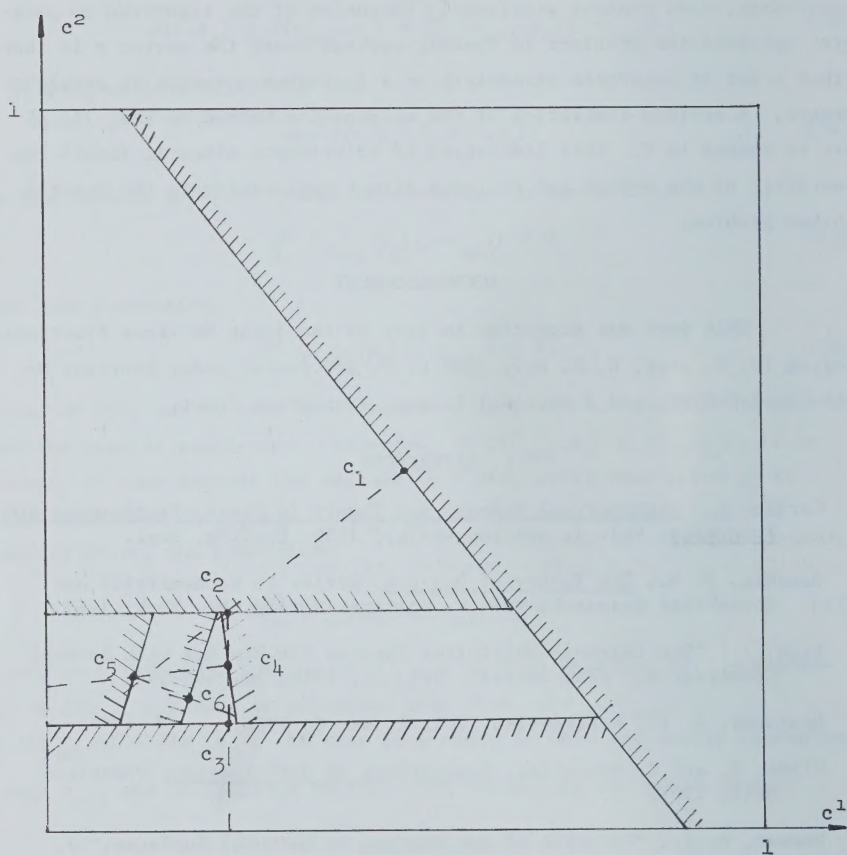


Fig. 1

A REEXAMINATION OF TERNARY SWITCHING THEORY

ANTHONY S. WOJCIK and GERNOT METZE
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois

ABSTRACT

This paper describes the logic development of ternary switching circuits and discusses some of the problems that arise both in the logical development and physical implementation of combinational and sequential ternary circuits.

INTRODUCTION

In recent years there has been an increased interest in the logic design of ternary switching circuits. Attempts have been made to point out the theoretical applicability of these circuits and to show how they may be physically realized. These investigations have led to interesting results. Yoeli and Rinen and also Eichelberger have proposed that ternary switching theory be used for designing hazard-free binary gating and series-parallel contact networks [1,2]. A Russian stored-program computer, the SET'UN, incorporates the ternary number system in the logic networks of its arithmetic unit to take advantage of the relative ease of representing both positive and negative numbers and the simplicity of performing non-biased round-off in the ternary representation [3]. It has been proposed that ternary logic redundancy be used in binary networks to improve their operating reliability [4], and it has been suggested that a ternary logic circuit be used instead of a double-rail carry indicator in computer arithmetic units. The use of ternary switching networks in the logic sequencing control points of a computer arithmetic control has also been investigated [5].

The purpose of this paper is to describe the logic development of ternary switching circuits and to discuss some of the problems that arise in both the logical development and physical implementation of combinational and sequential ternary circuits. We will parallel the construction of binary switching theory, pointing out the extensions to the ternary system and discussing some of the differences that exist.

TERNARY ALGEBRA AND FUNCTIONS

For our purposes, ternary switching theory can be based on a Post or lattice algebra with the operations of addition (+) and multiplication ($\cdot$) defined on the set $P = \{0, \frac{1}{2}, 1\}$. All ternary variables may assume any of the values of P .

We are able to represent ternary functions in terms of the ternary equivalents of the binary AND and OR gates by means of the correspondence

$$\text{OR} \equiv \max(x, y) = x + y$$

$$\text{AND} \equiv \min(x, y) = x \cdot y, \quad x, y \in P.$$

Ternary negation will be considered later. For a complete treatment of this ternary algebra, the reader is referred to the papers by Yoeli and Rosenfeld [6] and Abecassis [7].

There are 27 different unary (one-variable) functions from P to P. These functions are represented as

$$x^{ijk} = \begin{cases} i & \text{if } x = 0 \\ j & \text{if } x = \frac{1}{2} \\ k & \text{if } x = 1, \text{ for } i, j, k \in P. \end{cases}$$

Similar to the binary case, any ternary function of n-variables, $f(\vec{x}) \equiv f(x_1, \dots, x_n)$, can be represented in a canonical form

$$f(\vec{x}) = f_0(\vec{x}) + f_{\frac{1}{2}}(\vec{x}) + f_1(\vec{x})$$

in which the function f_0 attains only the value 0, $f_{\frac{1}{2}}$ may be either 0 or $\frac{1}{2}$, and f_1 is either 0 or 1 (see references [5, 6, 7]).

Ternary functions may be represented by a truth table. Table 1 represents a ternary 2-variable function. For notational convenience, the '.' is omitted between variables so that juxtaposition represents the 'min' operation. The canonical form of this function is:

$$\begin{aligned} f(x_1, x_2) &= f_0(x_1, x_2) + f_{\frac{1}{2}}(x_1, x_2) + f_1(x_1, x_2) \\ &= f_1(x_1, x_2) + f_{\frac{1}{2}}(x_1, x_2) \end{aligned}$$

$$\begin{aligned} \text{where } f_1(x_1, x_2) &= x_1^{100} x_2^{010} + x_1^{100} x_2^{001} + x_1^{010} x_2^{001} + x_1^{001} x_2^{001} \\ f_{\frac{1}{2}}(x_1, x_2) &= x_1^{0\frac{1}{2}0} x_2^{\frac{1}{2}00} + x_1^{00\frac{1}{2}} x_2^{\frac{1}{2}00} + x_1^{00\frac{1}{2}} x_2^{0\frac{1}{2}0} \end{aligned}$$

We then have

$$\begin{aligned} f(x_1, x_2) &= x_1^{100} x_2^{010} + x_1^{100} x_2^{001} + x_1^{010} x_2^{001} + x_1^{001} x_2^{001} \\ &\quad + x_1^{0\frac{1}{2}0} x_2^{\frac{1}{2}00} + x_1^{00\frac{1}{2}} x_2^{\frac{1}{2}00} + x_1^{00\frac{1}{2}} x_2^{0\frac{1}{2}0} \\ &= x_1^{100} \cdot (x_2^{010} + x_2^{001}) + (x_1^{010} + x_1^{001}) \cdot x_2^{001} \\ &\quad + (x_1^{0\frac{1}{2}0} + x_1^{00\frac{1}{2}}) \cdot x_2^{\frac{1}{2}00} + x_1^{00\frac{1}{2}} \cdot (x_2^{\frac{1}{2}00} + x_2^{0\frac{1}{2}0}) \\ &= x_1^{100} x_2^{011} + x_2^{001} + x_1^{0\frac{1}{2}\frac{1}{2}} x_2^{\frac{1}{2}00} + x_1^{00\frac{1}{2}} x_2^{\frac{1}{2}\frac{1}{2}0} \end{aligned}$$

Two possible extensions of the binary NOT operation to the ternary system are listed in Table 2. Operation C is known as cycling, while N stands for negation. These two operations reflect the distinction between the linear ordering and the phase shift or rotational ordering of the ternary values. Rotational ordering makes each of the ternary values adjacent, Figure 1a. Linear ordering does not allow the adjacency of 1 and 0, and a ternary variable whose value is 0(1) can change to the value 1(0) only by first attaining the value $\frac{1}{2}$, Figure 1b.

An examination of Table 2 reveals that if we impose the restriction of linear ordering on the ternary values, the N operation will preserve this linearity. If we allow the rotational scheme, then either the C or the N operations may be used.

The distinction between the linear and rotational orderings is not important if we are only interested in synchronous circuits. A change in a variable value from 1 to 0 is possible since the intermediate value of $\frac{1}{2}$ can be made to occur between clock pulses. Thus the presence of the $\frac{1}{2}$ -value need not be taken into account. The importance of this difference

in ordering is best exemplified by considering hazards in ternary asynchronous networks.

TERNARY MAPS

Although we can consider the concept of hazards from a purely algebraic point of view, it is useful to extend the binary Karnaugh map to the ternary system so that we may be able to represent hazards graphically. These maps will also be helpful in the design of ternary circuits.

In ternary switching theory, these maps are called T-maps [5]. Table 3 contains the T-map representing the function of Table 1

$$f(x_1, x_2) = x_1^{100} x_2^{011} + x_2^{001} + x_1^{012} x_2^{100} + x_1^{001} x_2^{120}.$$

The geometric form of the T-map (i.e., the heavy lines of Figure 2) is that of a lattice whose nodes correspond to ternary n-tuples and whose edges represent all possible adjacencies of n-tuples that differ in at most one component. Since the T-map does not allow the adjacency of the variable value 0 to the variable value 1, it is based on the linear, rather than on the rotational, ordering of the ternary variables. If the rotational ordering is to be used, the adjacency of 0 and 1 must be allowed, as indicated by the dotted lines in Figure 2.

FUNCTION SIMPLIFICATION

Each entry of the T-map must be a 0, $\frac{1}{2}$, 1, or d (don't care). A 0-entry is referred to as a 0-square, etc. We are concerned with sets of squares that contain only entries of the same kind and possible d-entries. With respect to the canonical form of the corresponding function, each square of the map is an n-tuple of coordinates represented by a term in the function which is:

for a 1-square: $x_1^{i_1 j_1 k_1} \dots x_n^{i_n j_n k_n}$

for a $\frac{1}{2}$ -square: $x_1^{\hat{i}_1 \hat{j}_1 \hat{k}_1} \dots x_n^{\hat{i}_n \hat{j}_n \hat{k}_n}$

for a 0-square: no term

for a d-square: the representation used for a 0-, $\frac{1}{2}$ -, or 1-square depending on how the d-square is to be interpreted,

where $i_\ell, j_\ell, k_\ell \in \{0, 1\}$ and $\hat{i}_\ell, \hat{j}_\ell, \hat{k}_\ell \in \{0, \frac{1}{2}\} \forall \ell = 1, \dots, n$.

The basic principle of the map method of functional simplification is the combination of squares into the largest possible sets. We may combine adjacent squares of the same kind, consider 1-squares as $\frac{1}{2}$ -squares for the formation of $\frac{1}{2}$ -sets, and allow a square to belong to more than one set. Complete details may be found in [5] and [7].

To illustrate the simplification procedure, consider the function given by Table 4, in which the letters in each square are for identification. The following combinations are possible:

B and C: $x_1^{011} x_2^{1/2 00}$

D and G: $x_1^{100} x_2^{011}$

G, H and I: x_2^{001}

C, F and I: $x_1^{00} x_2^{1/2}$

This results in the simplification

$$f(x_1, x_2) = X_1^{100} X_2^{011} + X_2^{001} + X_1^{0\frac{1}{2}} X_2^{\frac{1}{2}00} + X_1^{00\frac{1}{2}}.$$

We can illustrate the distinction between linear and rotational orderings in map simplification with Table 5. The function represented is

$$f(x_1, x_2) = X_1^{100} X_2^{100} + X_1^{010} X_2^{001} + X_1^{001} X_2^{100} + X_1^{\frac{1}{2}00} X_2^{0\frac{1}{2}0} + X_1^{0\frac{1}{2}0} X_2^{0\frac{1}{2}0} + X_1^{00\frac{1}{2}} X_2^{0\frac{1}{2}0}.$$

The heavy lines indicate sets of squares that may be combined in both ordering schemes. The dotted line indicates a 1-set that may be formed in the rotational but not in the linear ordering, for the linear ordering does not allow the x_1 variable to change from 0 to 1 and from 1 to 0. It is also apparent that Table 5 does not represent a legal ternary function in the linear ordering system since there are several possible sequences of output values that are not legal sequences in the linear scheme.

From this discussion, the function defined in Table 6 can be realized directly in the rotational but not in the linear ordering system because of the functional value changes from 0 to 1 and 1 to 0. Suppose we represent this function by the function $g(x)$ in Table 7. The values attained by $g(x)$ comply with the linear ordering scheme since in both the transitions from 1 to 0 and 0 to 1 the intermediate value $\frac{1}{2}$ is attained and not merely implied. In a network in which $g(x)$ is also an input to another part of the circuit, the $\frac{1}{2}$ value will occur as the output value changes from 0 to 1 or from 1 to 0 and must be recognized by the rest of the network.

The function $g(x)$ and others like it form a class of ternary functions that has not been thoroughly investigated. These functions may prove to be very useful in the design of ternary networks, as a later example will show.

HAZARDS IN TERNARY NETWORKS

Following the development of Abecassis [7], we consider the problem of hazards in asynchronous ternary circuits. At first the restriction of the linear ordering of the variables will not be imposed.

Let us consider an arbitrary network and two adjacent input vectors $\vec{a} \equiv (a_1, \dots, a_i, \dots, a_n)$ and $\vec{b} \equiv (a_1, \dots, b_i, \dots, a_n)$ such that

$$f(\vec{a}) = f(\vec{b}).$$

Changing the input from $\vec{a}$ to $\vec{b}$ causes a static hazard if the output of the network has a transient value different from $f(\vec{a})$ before the network becomes stable.

For example, consider the function represented by Table 8,

$f(x_1, x_2) = X_1^{100} X_2^{110} + X_1^{011} X_2^{100} + X_2^{00\frac{1}{2}} + X_1^{\frac{1}{2}0\frac{1}{2}}.$ If we start with an initial input state of $x_1, x_2 = 00$ and allow the x_1 input to change to 1, which is allowable in the rotational ordering system, the value of f should remain 1. However, if the circuit realization of f is the one suggested by the $\frac{1}{2}$ -set and 1-set covers in Table 8, then a transient output of $\frac{1}{2}$ could occur, depending on the magnitude of the line delays.

If it is the case that

$$f(\vec{a}) \neq f(\vec{b}),$$

then a change of input from $\vec{a}$ to $\vec{b}$ results in a dynamic hazard if the output transition from $f(\vec{a})$ to $f(\vec{b})$ does not occur directly.

To illustrate a dynamic hazard, we consider Table 9 which represents the same function as Table 8 but has a different collection of covering sets. In this case,

$$f(x_1, x_2) = x_2^{100} + x_1^{100} x_2^{110} + x_1^{00\frac{1}{2}} + x_2^{00\frac{1}{2}}.$$

With the suggested circuit realization for f , the output sequences 1, 0, $\frac{1}{2}$ might occur as the inputs change from $x_1 x_2 = 0\frac{1}{2}$ to $x_1 x_2 = 01$.

These examples only give a glimpse of the types of hazards that can occur in ternary networks. Abecassis' paper goes into greater detail. We state his final theorem:

THEOREM (Abecassis): Temporary and erroneous outputs from networks without feedback can be eliminated by the following rules:

- (1) By insuring that
 - a) each pair of adjacent 1-squares is covered by at least one 1-set of the network;
 - b) each pair of adjacent $\frac{1}{2}$ -squares is covered by at least one $\frac{1}{2}$ -set of the network;
 - c) each pair of adjacent 1-square and $\frac{1}{2}$ -square combinations is covered by at least one $\frac{1}{2}$ -set of the network;
 - d) no pair of adjacent 1-square and 0-square combinations appears in the Karnaugh map [i.e., T-map] representing f .
- (2) By restricting changes of inputs to a single variable changing at any one time, a change from 0 to 1 or 1 to 0 not being permitted.
- (3) By waiting a sufficient length of time between input changes for all elements to become stable.

It is evident that if we want to obtain hazard-free ternary combinational circuits, the linear ordering must be imposed.

TABULAR MINIMIZATION PROCEDURES

Several authors have extended the McCluskey tabular method for minimizing binary combinational circuits to the ternary case, but they have not considered the problem of hazards [6,8,10]. As a result, the network realizations obtained by their procedures use non-hazard-free unary functions. Presently, we are developing a tabular method for obtaining hazard-free realizations of ternary functions. The method will determine whether or not the function to be synthesized is inherently hazard-free. If it is, a hazard-free realization results; if not, a non-hazard-free solution can be obtained.

SEQUENTIAL CIRCUITS

Some authors have dismissed the topic of ternary sequential circuits with the observation that this is an obvious extension of the

binary theory. Indeed, this is true if we have a ternary memory element to use. The design and construction of tristable core or film memories for use as bulk memory has been investigated [9,10,11]. Hence we have physical devices that can be used, but we must overcome our conceptual difficulty of being unable to ascribe a logical significance to the $\frac{1}{2}$ -level signal. In a ternary analogue of the binary flip-flop, the significance of a $\frac{1}{2}$ set signal or $\frac{1}{2}$ reset signal, the information stored when these signals are present, and the memory state of such a device all must be determined. Some proposed ternary flip-flop devices only utilize a few of the total possible number of input combinations that can occur. These allowed combinations are not chosen because they provide some special information. The other combinations are relegated to the 'don't occur' category of inputs because they are logically obscure.

TERNARY MEMORY DEVICES

One possible memory device is a ternary extension of the binary flip-flop. In Figure 3 we interpret the OR gate as the ternary min operation and the NOT gate as the ternary N operation. Figure 4 indicates all the legal transitions that may occur under the linear ordering of the ternary values if the input variables S and R begin in the SR = 10 state. Analysis of possible input sequences reveals that the SR = 00 state acts as the memory state with $T_1 F_1 = 01, 10$, or $\frac{1}{2}\frac{1}{2}$ as stable states. Starting in this memory state, the following is evident:

SET SEQUENCE:	SR:00 - $\frac{1}{2}0$ ----- 10 ----- $\frac{1}{2}0$ - 00
	$T_1 F_1$:01 - 01 - $\frac{1}{2}1$ - $\frac{1}{2}\frac{1}{2}$ - $1\frac{1}{2}$ - 10 - 10
RESET SEQUENCE:	SR:00 - $0\frac{1}{2}$ ----- 01 ----- $0\frac{1}{2}$ - 00
	$T_1 F_1$:10 - 10 - $1\frac{1}{2}$ - $\frac{1}{2}\frac{1}{2}$ - $\frac{1}{2}1$ - 01 - 01
$\frac{1}{2}$ -SET SEQUENCES:	SR:00 - $0\frac{1}{2}$ ----- 00
	$T_1 F_1$:10 - 10 - $1\frac{1}{2}$ - $\frac{1}{2}\frac{1}{2}$
	SR:00 - $\frac{1}{2}0$ ----- 00
	$T_1 F_1$:01 - 01 - $\frac{1}{2}1$ - $\frac{1}{2}\frac{1}{2}$.

Although all possible input combinations are not used, the $\frac{1}{2}\frac{1}{2}$ output state can be logically interpreted as a third memory state or as a state indicating that the network is waiting to change to either the 0 or the 1 state.

As another possible device, we replace the N element with the ternary function defined in Table 10. We then consider the ternary flip-flop of Figure 5. Table 11a lists a Set sequence for the device of Figure 3 while Table 11b indicates a Set sequence for Figure 5. In Table 11a, when $T_1 = 1$ at point h, we do not know if the OR gate has the output $T_1 = 1$ because of the T_2 input or the S input. We are unable to determine if the T_2 input is stuck at 0 without first removing the S = 1 input. If the circuit is malfunctioning, the information that was to be stored is lost. Table 11b shows that when $T_1 = 1$ at point h, the T_1 output is at the 1 level only because the T_2 input is 1. Therefore, the Set signal may be removed when $T_1 = 1$ without loss of information. In this scheme we can take the output value of the circuit from T_1 since it is 'the last moving point' of the network. An analogous argument reveals similar results with the reset line (R) and the circuit output taken from F_1 .

CONCLUSION

We have presented a brief development of ternary switching theory. Analysis of both combinational and sequential ternary circuits has indicated some of the similarities to and differences from binary networks. The topics of hazards, variable value ordering schemes, and ternary memory devices have been examined and pointed out to be areas of ternary switching theory that need further development. We feel that ternary networks and applications of many-valued logics to computer logic design are sources of interest that will continue to be profitably investigated.

ACKNOWLEDGMENT

This work has supported in part by the National Science Foundation under Grant NSF-GK 1663.

REFERENCES

- [1] Yoeli, M. and S. Rinen, "Application of Ternary Algebra to the Study of Static Hazards," JACM, vol. 11, pp.84-97, January, 1964.
- [2] Eichelberger, E.B., "Hazard Detection in Combinational and Sequential Switching Circuits," IBM J. Res. and Dev., vol.9, pp.90-99, March, 1965.
- [3] Zhogolev, Y.A., "The Order Code and an Interpretative System for the SET'UN Computer," USSR Comp. Math. and Math. Phys., vol.3, Oxford: Pergamon, pp.563-578, 1962.
- [4] Varshavskii, V.I., "Ternary Majority Logic," Automatika i Telemekhanika, vol.25, pp.673-684, May, 1964.
- [5] Wojcik, A.S., "An Application of Ternary Logic to Logic Sequencing," University of Illinois, Coordinated Science Laboratory Report R-388, July, 1968.
- [6] Yoeli, M. and G. Rosenfeld, "Logical Design of Ternary Switching Circuits," IEEE Transactions on Electronic Computers, pp.19-29, February, 1965.
- [7] Abecassis, L., "A Study of Hazards in Ternary Switching Circuits," University of Illinois, Coordinated Science Laboratory Report R-310, August, 1966.
- [8] Merrill, R.D., Jr., "A Tabular Minimization Procedure for Ternary Switching Functions," IEEE Transactions on Electronic Computers, pp.578-585, August, 1966.
- [9] Santos, J., H. Arango, and M. Pascual, "A Ternary Storage Element Using a Conventional Ferrite Core," IEEE Transactions on Electronic Computers, p.248, April, 1965.

- [10] Santos, J. and M. Arango, "On the Analysis and Synthesis of Three-Valued Digital Systems," AFIPS Conference Proceedings, Vol.25, Spring Joint Computer Conference, 1964.
- [11] Schauer, R.F., R.M. Steward, Jr., A.V. Pohm, and A.A. Read, "Some Applications of Magnetic Film Parametrons as Logical Devices," IRE Transactions on Electronic Computers, pp.315-320, September, 1960.
-

TABLE 1

x_1	x_2	f
0	0	0
0	$\frac{1}{2}$	1
0	1	1
$\frac{1}{2}$	0	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$	0
$\frac{1}{2}$	1	1
1	0	$\frac{1}{2}$
1	$\frac{1}{2}$	$\frac{1}{2}$
1	1	1

TABLE 2

X	C	N
0	$\frac{1}{2}$	1
$\frac{1}{2}$	1	$\frac{1}{2}$
1	0	0

TABLE 3

x_1	x_2	0	$\frac{1}{2}$	1
0		0	1	1
$\frac{1}{2}$		$\frac{1}{2}$	0	1
1		$\frac{1}{2}$	$\frac{1}{2}$	1

T-MAP

FIGURE 2

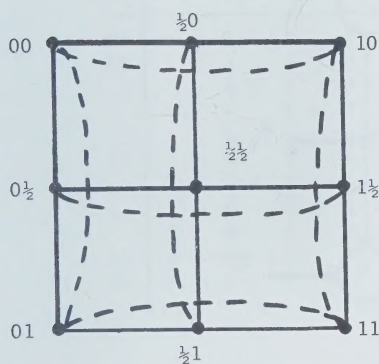


FIGURE 1

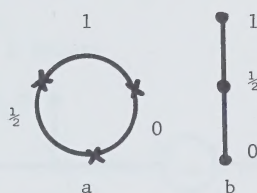


TABLE 4

x_1	x_2	0	$\frac{1}{2}$	1
0		0 A	1 D	1 G
$\frac{1}{2}$		$\frac{1}{2}$ B	0 E	1 H
1		$\frac{1}{2}$ C	$\frac{1}{2}$ F	1 I

TABLE 5

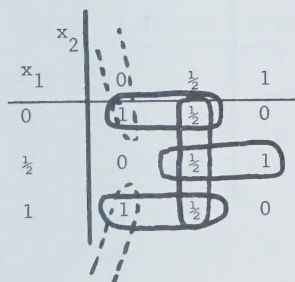


TABLE 6

x	$f(x)$
0	1
$\frac{1}{2}$	0
1	1

TABLE 7

x	$g(x)$
0	1
$\frac{1}{2}$	$\frac{1}{2}$
1	1

TABLE 8

x_2	0	$\frac{1}{2}$	1
0	1	1	$\frac{1}{2}$
$\frac{1}{2}$	1	0	$\frac{1}{2}$
1	1	$\frac{1}{2}$	$\frac{1}{2}$

TABLE 9

x_2	0	$\frac{1}{2}$	1
0	1	1	$\frac{1}{2}$
$\frac{1}{2}$	1	0	$\frac{1}{2}$
1	1	$\frac{1}{2}$	$\frac{1}{2}$

FIGURE 3

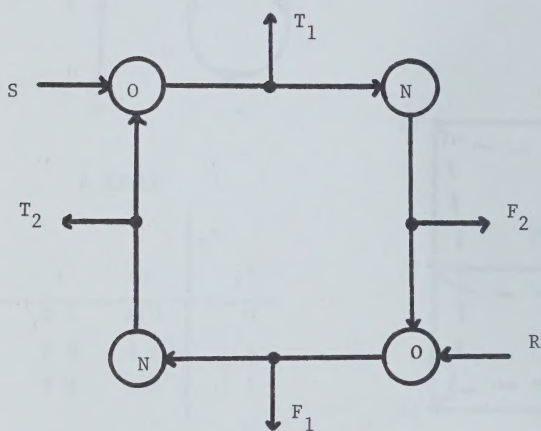
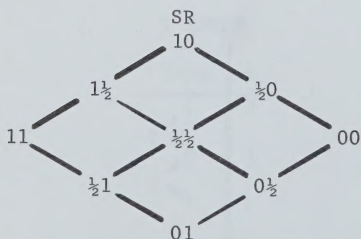


FIGURE 4



INPUT	OUTPUT
0	1
$0 - \frac{1}{2}$	$1 - \frac{1}{2} - 0$
$\frac{1}{2}$	0
1	0

FIGURE 5

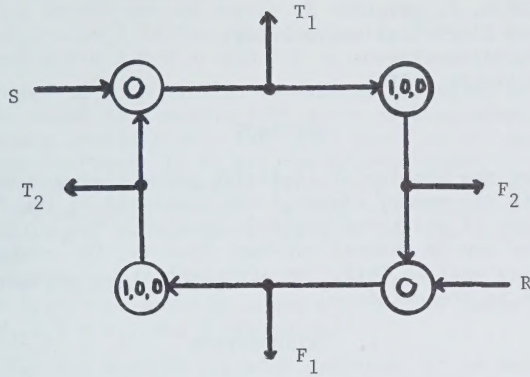


TABLE 11

SR	T_1	F_2	F_1	T_2
00	0	1	1	0
$\frac{1}{2}0$	$\frac{1}{2}$ a	1	1	0
	$\frac{1}{2}$	$\frac{1}{2}$ b	1	0
	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$ c	0
	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$ d
10	1 e	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
	1	0 f	$\frac{1}{2}$	$\frac{1}{2}$
	1	0	0 g	$\frac{1}{2}$
	1	0	0	1 h
$\frac{1}{2}0$	1	0	0	1
00	1	0	0	1
a				

SR	T_1	F_2	F_1	T_2
00	0	1	1	0
$\frac{1}{2}0$	$\frac{1}{2}$ a	1	1	0
	$\frac{1}{2}$	$\frac{1}{2}$ b	1	0
	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$ c	0
	$\frac{1}{2}$	0 d	$\frac{1}{2}$	0
	$\frac{1}{2}$	0	0 e	0
	$\frac{1}{2}$	0	0	$\frac{1}{2}$ f
	$\frac{1}{2}$	0	0	1 g
	1 h	0	0	1
00	1	0	0	1
	1	0	0	1
b				

S. S. YAU AND G. F. DEPALMA
Department of Electrical Engineering
Northwestern University
Evanston, Illinois 60201

ABSTRACT

In this paper the problem of realizing sequential machines with a structure in which the memory elements are connected in the form consisting of a minimum number of shift-registers is studied. The shift-registers in these realizations are in general two-way, that is, the content can be shifted to right as well as left. An efficient state assignment method for such realizations is presented.

I. INTRODUCTION

From economical and reliability points of view, it is desirable to realize synchronous sequential machines with a structure in which the memory elements are connected in the form consisting of a minimum number of shift registers. This problem has been studied by a number of authors [1]-[6], but the shift registers used in their realizations are those, called one-way shift registers, which can shift their contents only in one direction. In this paper, we shall consider the use of two-way shift registers in realizing synchronous sequential machines. A 3-bit two-way shift register is schematically shown in Fig. 1(a). The input h_1 or h_2 supplies the new bit to be shifted in when the content is shifted to the right or left respectively. The direction of the shift is controlled by the input g , with the convention $g=0$ for a right shift and $g=1$ for a left shift.¹ Without loss of generality, we can connect h_1 and h_2 together as shown in Fig. 1(b). The state graphs of a 3-bit one-way shift register and a 3-bit two-way shift register are shown in Figs. 2(a) and 2(b) respectively. In Fig. 2(b), the first bit represents the h input and the second bit represents the g input. It is obvious that when the switching function g of a two-way shift register is the constant 0 or 1, this two-way shift register acts as a one-way shift register and consequently can be replaced by a one-way shift register. Therefore, one-way shift registers can be considered as a special case of two-way shift registers. In this paper, we shall present a general state assignment method for realizing a synchronous sequential machine using a minimum number of shift registers (two-way or one-way). The method is based on matrix algebra and can be easily programmed. The general structure of such a realization is shown in Fig. 3. An example will be given to show that a single two-way shift-register realizable sequential machine may not be realizable using two one-way shift registers.

II. BASIC THEOREMS

In this section, we shall lay out the foundation for our general shift-

^{*}The work reported here was supported in part by the Air Force Office of Scientific Research, Office of Aerospace Research, United States Air Force, under AFOSR Grant No. AF-AFOSR-1292-67.

¹Such a two-way shift register has been manufactured in a single integrated circuit package by Fairchild. All or part of it can be used, and it can be connected to other similar units to produce longer shift registers.

register realization technique for synchronous sequential machines.

Definition 1: A sequential machine is a 5-tuple $M = (S, I, W, f, g)$, where $S = \{s_1, s_2, \dots, s_n\}$ is the set of states, $I = \{i_1, i_2, \dots, i_v\}$ is the set of inputs, $W = \{w_1, w_2, \dots, w_u\}$ is the set of outputs, $f: S \times I \rightarrow S$ is the next-state function and $g: S \times I \rightarrow W$ or $g: S \rightarrow W$ is the output function.

The state behavior of a machine is usually specified by its state transition table, which is a matrix with a row corresponding to a state and a column to an input combination. The (j, k) entry in the matrix is the next state when the present state is s_j and the present input is i_k . As it is always possible to realize the output function g from the state and input variables by a combinational circuit, we will only consider the state behavior of M . Therefore, the output alphabet W and the output function g will be neglected.

Definition 2: The skeleton matrix of a sequential machine M with n states is a matrix $E_M = [e_{ij}^{(m)}]$ of order n , where $e_{ij}^{(m)} = 1$ if there exists an input i_k such that $f(s_i, i_k) = s_j$, and 0 otherwise.

Because of the way a shift register operates, it is easy to generate its skeleton matrix. For a left shift with a 0 or 1 entry, the binary code is multiplied by 2 or multiplied by 2 and added by 1. In both cases the multiplication is modulo 2^l , where l is the length of the register. For the right shift, the binary code is divided by 2 (and added by 2^{l-1} if a 1 is entered). The division is in the integer mode, that is, the fractional portion is eliminated. These formulas are used in programming the algorithm to generate the skeleton matrix.

A binary code C_ℓ is the set of all possible sequences of ℓ binary bits, where ℓ is a positive integer called the length of C_ℓ . Each sequence in C_ℓ is called a code word of C_ℓ . A coding on a set $S = \{s_1, s_2, \dots, s_n\}$ is an onto and single-valued function $\theta: C_\ell \rightarrow S$, where $C_\ell \subseteq C_\ell$.

Definition 3: Let θ be a many-to-one mapping from a set $T = \{t_1, t_2, \dots, t_m\}$ onto a set $S = \{s_1, s_2, \dots, s_n\}$. Then, the coding matrix for θ is a matrix $\underline{A} = [a_{ij}]$ of order $m \times n$, where $a_{ij} = 1$ if t_i is mapped to s_j , and 0 otherwise.

Since the mapping is many-to-one and onto, $\underline{A}$ satisfies the following conditions:

$$\sum_{i=1}^n a_{ij} \geq 1, \quad j = 1, 2, \dots, n, \quad (1)$$

$$\sum_{j=1}^m a_{ij} \leq 1, \quad i = 1, 2, \dots, m. \quad (2)$$

Furthermore, it is easy to show that for a matrix $\underline{A}$ of order $m \times n$ consisting of the entries 1 or 0 satisfying (1) and (2), a many-to-one and onto mapping can be found such that $\underline{A}$ is its coding matrix. Hence, we have the following lemma:

Lemma 1: A matrix $\underline{A} = [a_{ij}]$ of order $m \times n$ is a coding matrix if and only if it consists of entries only 1 or 0 and satisfies (1) and (2).

Now, we shall prove the following lemma:

Lemma 2: If $\underline{A}$ and $\underline{B}$ are coding matrices of orders $n \times p$ and $p \times m$ respectively, then the product $\underline{A} \underline{B}$ is also a coding matrix.

Proof: Let $\underline{C} = \underline{A} \underline{B} = [c_{ij}]$, where $\underline{A} = [a_{ij}]$ and $\underline{B} = [b_{ij}]$ are the two given coding matrices. Since $c_{ij} = \sum_{k=1}^p a_{ik} b_{kj}$ and $\sum_{k=1}^p a_{ik} \leq 1$, $c_{ij} = 0$ or 1.

Because $\sum_{i=1}^n a_{ik} \geq 1$ and $\sum_{k=1}^p b_{kj} \geq 1$, we have

$$\sum_{i=1}^n c_{ij} = \sum_{i=1}^n \sum_{k=1}^p a_{ik} b_{kj} = \sum_{k=1}^p [b_{kj} \sum_{i=1}^n a_{ik}] \geq \sum_{k=1}^p b_{kj} \geq 1. \quad (3)$$

Furthermore,

$$\sum_{j=1}^m c_{ij} = \sum_{k=1}^p [a_{ik} \sum_{j=1}^m b_{kj}] \leq \sum_{k=1}^p a_{ik} \leq 1. \quad (4)$$

It follows from Lemma 1 that $\underline{C}$ is a coding matrix.

Definition 4: A machine $M = (S, I, f)$, where $S = \{s_1, s_2, \dots, s_n\}$, is said to be embedded in a machine $R = (T, I', f')$, where $T = \{t_1, t_2, \dots, t_m\}$, if there exists an onto and single-valued function $\Psi: T' \rightarrow S$, where $T' \subseteq T$, such that if $\Psi(t_k) = s_q$, then $\Psi[f'(t_k, I')] \supseteq f(s_q, I)$, where $f(s_q, I) = \{s_j | s_j = f(s_q, i_k), i_k \in I\}$.

It is obvious that the function Ψ can be represented by a coding matrix $\underline{A}$.

Theorem 1: A machine $M = (S, I, f)$ can be embedded in a machine $R = (T, I', f')$ if and only if there exists a coding matrix $\underline{A}$ from T to S such that

$$\underline{E}_R \underline{A} - \underline{A} \underline{E}_M \geq \underline{0}, \quad (5)$$

where $\underline{E}_R$ and $\underline{E}_M$ are the skeleton matrices of R and M respectively.

Proof: Assume there exists a function Ψ which embeds M in R . Define the matrix $\underline{A} = [a_{ij}]$, where

$$a_{ij} = \begin{cases} 1 & \text{if } \Psi(t_i) = s_j \\ 0 & \text{otherwise.} \end{cases}$$

Since Ψ is an onto and single-valued function, $\underline{A}$ satisfies (1) and (2), and hence is a coding matrix. Let $e_{ij}^{(r)}$ and $e_{ij}^{(m)}$ be the (i, j) entries of $\underline{E}_R$ and $\underline{E}_M$ respectively. Let $\underline{H} = [h_{ij}]$ be the matrix $\underline{E}_R \underline{A} - \underline{A} \underline{E}_M$. It is seen that h_{ij} can be negative only if

$$\sum_{k=1}^m e_{ik}^{(r)} a_{kj} < \sum_{k=1}^n a_{ik} e_{kj}^{(m)}. \quad (6)$$

Now, we would like to show that the relation (6) cannot hold. It is noted

$\sum_{k=1}^n a_{ik} e_{kj}^{(m)} \leq 1$, and that equality holds if there exists a p such that $a_{ip} = e_{pj}^{(m)} = 1$. The condition $a_{ip} = 1$ implies $\Psi(t_i) = s_p$, and $e_{pj}^{(m)} = 1$ implies $s_j \in f(s_p, I)$. Since Ψ embeds M in R ,

$$\Psi[f'(t_i, I')] \supseteq f(s_p, I). \quad (7)$$

In other words, there exists a state $t_w \in f'(t_i, I')$ such that $\Psi(t_w) = s_j$.

This implies that $e_{iw}^{(r)} = a_{wj} = 1$, and consequently $\sum_{k=1}^m e_{ik}^{(r)} a_{kj} \geq 1$. Hence, (5) is satisfied.

Conversely, assume that there is a coding matrix $\underline{A}$ satisfying (5). Then, $\underline{A}$ defines a function $\Psi: T' \rightarrow S$, where $T' \subseteq T$. Because $\underline{A}$ satisfies (1) and (2), Ψ is onto and single-valued. Now, we have to show that if $\Psi(t_i) = s_j$, $\Psi[f'(t_i, I')] \supseteq f(s_j, I)$. Let $s_k \in f(s_j, I)$. Then, $e_{jk}^{(m)} = 1$ and $a_{ij} = 1$. This implies that $\sum_{p=1}^n a_{ip} e_{pk}^{(m)} = 1$. It follows from (5) that

$\sum_{q=1}^n e_{iq}^{(r)} a_{qk} \geq 1$. This implies that there exists a t_d such that

$t_d \in f'(t_i, I')$ and $\Psi(t_d) = s_k$. Consequently, we have

$$s_k = \Psi(t_d) \in \Psi[f'(t_i, I')]. \quad (8)$$

This implies that every $s_k \in f(s_j, I)$ is in $\Psi[f'(t_i, I')]$, and hence

$$f(s_j, I) \subseteq \Psi[f'(t_i, I')]. \quad (9)$$

It follows from Definition 4 that the function Ψ embeds M in R . This completes the proof of the theorem.

The following two theorems can easily be shown:

Theorem 2: If a machine M can be embedded in a machine R , then all machines in the set $G(M)$, where

$$G(M) = \{M' | \underline{E}_M' \leq \underline{P}^{-1} \underline{E}_M \underline{P}\} \quad (10)$$

and $\underline{P}$ is a permutation matrix, can be embedded in R .

Theorem 3: If a machine M_1 is embedded in a machine M_2 using the coding matrix $\underline{A}_1$, and if M_2 is embedded in a machine M_3 using the coding matrix $\underline{A}_2$, then M_1 can be embedded in M_3 using the coding matrix $\underline{A}_3 = \underline{A}_2 \underline{A}_1$.

Definition 5: If θ_ℓ is a coding on a machine M , then $\bar{\theta}_\ell$ denotes the coding with each code word of θ_ℓ replaced by its 1's complement, θ'_ℓ denotes the coding with each code word of θ_ℓ replaced by itself written in reverse order, and $\bar{\theta}'_\ell$ denotes the combination of complementing and reversing each code word of θ_ℓ .

To simplify the notation, each state of a shift register will be labeled t_i , where i is equal to the equivalent decimal number of the code word plus 1. Thus, $t_1 = 00...00$, $t_2 = 00...01, \dots, t_{2^\ell} = 11...11$. It is easy to show the following theorems:

Theorem 4: If θ_ℓ is a coding embedding a machine M into a one-way shift register of length ℓ , then $\bar{\theta}_\ell$ is also.

Theorem 5: If θ_ℓ is a coding embedding a machine M into a two-way shift register of length ℓ , then θ_ℓ , $\bar{\theta}_\ell$, and $\bar{\theta}'_\ell$ are also.

Because of the above theorems, we make the following definitions:

Definition 6: Two coded states, t_i and t_j , of a one-way shift register are similar, denoted by $t_i \simeq t_j$, if and only if either $i = j$ or t_i and t_j are the 1's complement of each other. The similarity relation partitions the set T of coded states into a number of similarity classes.

Definition 7: Two coded states, t_i and t_j , of a two-way shift register are similar, denoted by $t_i \simeq t_j$, if and only if one of the following conditions is satisfied: (1) $i = j$; (2) t_i and t_j are the 1's complement of each other; (3) t_i and t_j are the reverse of each other; and (4) the combination of (2) and (3). The similarity relation partitions the set T of coded states into a number of similarity classes.

For example, 0110 and 1001 are similar in both the one-way and two-way shift registers. However, 0111 and 1110 are similar in the two-way shift register only.

III. SINGLE SHIFT-REGISTER REALIZATIONS

In this section we shall present an algorithm for single shift-register realizations of sequential machines based on embedding the given machine into a shift register. The shift register may be one-way or two-way.

Let the given machine $M = (S, I, f)$, where $S = \{s_1, s_2, \dots, s_n\}$, and the shift register be $R = (T, I', f')$, where $T = \{t_1, t_2, \dots, t_{2^\ell}\}$, and $n \leq 2^\ell$. The state assignment algorithm is formulated as follows:

1) Let $\ell = \lceil \log_2 n \rceil$, where $\lceil x \rceil$ denotes the smallest integer not less than x . Set a limit, d , on the maximum length of the shift register over ℓ .
 2) Generate the skeleton matrix, E_M , of machine M .
 3) Generate the skeleton matrix, E_R , of a shift register of length ℓ . Let $\{t_{i_1}, t_{i_2}, \dots, t_{i_r}\}$ be a set consisting of one member of each similarity class of T . Let U be a storage of matrices of order $2^\ell \times n$. Let B_{ij} be the matrix of order $2^\ell \times n$, whose entries are given by $b_{hk} = 1$ if $h = i$ and $k = j$ and 0 otherwise. Put the following matrices in U : $B_{i_1 1}, B_{i_2 1}, \dots, B_{i_r 1}$, and go to Step 4).

4) Take a matrix, say A , from U . Compute the matrix $H = E_R A - A E_M$. If H has a negative entry, go to Step 6). Otherwise, go to Step 7). If U was empty, go to Step 5).

5) If there were solutions generated, STOP. If not, set $\ell = \ell + 1$. If ℓ is now greater than $\lceil \log_2 n \rceil + d$, STOP. Otherwise, go to Step 3).

6) Pick a negative entry of H , say h_{ij} . Compute the intersection of the following two sets:

$$D_1 = \{k \mid \sum_{u=1}^h a_{ku} = 0\}$$

$$D_2 = \{k \mid e_{ik}^{(r)} = 1\}$$

If $D_1 \cap D_2 = \emptyset$, the null set, discard the matrix A , and go to Step 4). If $D_1 \cap D_2 = \{k_1, k_2, \dots, k_v\} \neq \emptyset$, add the following matrices to U : $A+B_{k_1, j}$; $A+B_{k_2, j}$; ...; $A+B_{k_v, j}$; and go to Step 4).

7) Let $d_j = \sum_{i=1}^{2^\ell} a_{ij}$. If d_j is greater than zero for all $j = 1, \dots, m$, then A is a solution. If $d_j = 0$ for some j , say $j = j_1, j_2, \dots, j_w$, and if $w > 2^\ell - \sum_{j=1}^n d_j$, discard the matrix A and go to Step 4). If $d_j = 0$ for some j , say $j = j_1, j_2, \dots, j_w$, and if $w \leq 2^\ell - \sum_{j=1}^n d_j$, add the following matrices to U : $A+B_{k_1, j_1}$; $A+B_{k_2, j_1}$; ...; $A+B_{k_p, j_1}$, where $\{k_1, k_2, \dots, k_p\} = D_1 = \{k \mid \sum_{u=1}^n a_{ku} = 0\}$. Go to Step 4).

If the machine M is strongly connected, that is, for every pair of states (s_i, s_j) there exists an input sequence $i_{j_1}, i_{j_2}, \dots, i_{j_r}$, such that

$$s_j = f(f(\dots f(f(s_i, i_{j_1}), i_{j_2}) \dots), i_{j_r}),$$

then Step 7) of the algorithm can be shortened to:

7) A is a solution, go to Step 4) for other solutions.

It should be noted that the algorithm will generate all the codings for single shift-register realization if Theorem 4 or 5 is applied to the solutions.

The algorithm has been programmed in FORTRAN on a CDC 6400 and the machine whose flow table is given in Fig. 4 was tried. The codings shown in Fig. 5(a) and two other codings were obtained in 105 seconds.

Once a suitable state assignment is chosen, the switching functions for h and g can be found by the following algorithm:

1) Generate the coded state transition table as follows: List all code words of C_ℓ down the left margin and all inputs across the top. To fill in the rows, we enter don't-cares (-) if that code word was not assigned. If the code word was assigned, we go to the transition table and see if that transition is specified. If it is don't-care, we enter that. If not, we enter the code word of the next state if there is only one. If the next state has more than one code word, we enter those which can be

obtained by (i) shifting the present state code word to the right and entering a new bit at the left, or (ii) shifting the present state code word to the left and entering a new bit at the right.

2) The Karnaugh maps for g and h are obtained as follows: (If more than one entry appears in the coded transition table, treat each entry separately.) If the next-state code word is obtained by (i) above but not by (ii), $g=0$ and $h=y_1$. If the next-state code word was obtained by (ii) and not by (i), $g=1$ and $h=y_1$. If the next-state code word could be obtained by either (i) or (ii), enter as follows: If $y_1=y_1$, g is don't care and $h=y_1$. If $y_1 \neq y_1$ make two separate entries of $g=1$ and $h=y_1$, and $g=0$ and $h=y_1$. If the next-state code word is don't-care, enter this in the Karnaugh map.

When there are multiple entries in the Karnaugh map, any one may be chosen for h as long as the corresponding one for g is chosen. The coded state transition table and Karnaugh maps for the machine of Fig. 4 are shown in Figs. 5(a) and 5(b) respectively. The resultant switching functions g and h are given by the following equations:

$$g = x_1 + x_2 y_2 (\bar{y}_3 y_4 + \bar{y}_1 y_3 \bar{y}_4)$$

$$h = x_1 (\bar{y}_1 + y_2 + y_3 y_4) + x_2 (y_2 \bar{y}_3 \bar{y}_4) + \bar{x}_2 (\bar{y}_2 + y_3).$$

It can be shown that the machine is not realizable by using two one-way shift registers.

IV. MULTIPLE SHIFT-REGISTER REALIZATIONS

Definition 8: Given two matrices, $\underline{A} = [a_{ij}]$ and $\underline{B} = [b_{ij}]$, the tensor product of $\underline{A}$ and $\underline{B}$, denoted by $\underline{A} \otimes \underline{B}$, is the matrix

$$\begin{bmatrix} a_{11}\underline{B} & a_{12}\underline{B} & \dots & a_{1n}\underline{B} \\ a_{21}\underline{B} & a_{22}\underline{B} & \dots & a_{2n}\underline{B} \\ \vdots & \vdots & & \vdots \\ a_{m1}\underline{B} & a_{m2}\underline{B} & \dots & a_{mn}\underline{B} \end{bmatrix},$$

where $\underline{A}$ and $\underline{B}$ are of order $m \times n$ and $r \times s$ respectively. Thus, $\underline{A} \otimes \underline{B}$ is of order $mr \times ns$.

Theorem 6: If $M_1, M_2, \dots, M_k$ are sequential machines with the state sets $S_1, S_2, \dots, S_k$, the input sets $I_1, I_2, \dots, I_k$, and the next-state functions $f_1, f_2, \dots, f_k$, then the machine $M = M_1 \times M_2 \times \dots \times M_k$ with the state set $S = S_1 \times S_2 \times \dots \times S_k$, the input alphabet $I = I_1 \times I_2 \times \dots \times I_k$, and the next-state function $f[(s_{1j_1}, s_{2j_2}, \dots, s_{kj_k}), (i_{1i_1}, i_{2i_2}, \dots, i_{ki_k})] = [f_1(s_{1j_1}, i_{1i_1}), f_2(s_{2j_2}, i_{2i_2}), \dots, f_k(s_{kj_k}, i_{ki_k})]$ has the skeleton matrix

$\underline{E}_M = \underline{E}_{M_1} \otimes \underline{E}_{M_2} \otimes \dots \otimes \underline{E}_{M_k}$, where $\times$ denotes the Cartesian product.

Proof: By induction on k. Assume $k=2$. Then, for a state s_i of M, where $s_i = (s_{1i_1}, s_{2i_2})$, then the next-state set $f(s_i, I) = [f_1(s_{1i_1}, I_1),$

$f_2(s_{2i_2}, I_2)]$. Numbering the states of M with the convention that $s_1 = (s_{1,1}, s_{2,1})$, $s_2 = (s_{1,1}, s_{2,2}), \dots, s_n = (s_{1,1}, s_{2,n})$, $s_{n+1} = (s_{1,2}, s_{2,1}), \dots, s_{2n} = (s_{1,2}, s_{2,n}), \dots, s_{mn} = (s_{1,m}, s_{2,n})$, where the states of M_1 are $s_{1,1}, s_{1,2}, \dots, s_{1,m}$, and the states of M_2 are $s_{2,1}, s_{2,2}, \dots, s_{2,n}$. In other words, state $s_i = (s_{1,p}, s_{2,r})$, where $i = (p-1)n + r$ for $1 \leq r \leq n$. Then, the entries of the i th row of $\underline{E}_{M_1} \otimes \underline{E}_{M_2}$ are $a_{p1}b_{r1}, a_{p1}b_{r2}, \dots, a_{p1}b_{rn}$,

$a_{p2^{b_{r1}}}, a_{p2^{b_{r2}}}, \dots, a_{p2^{b_{rn}}}$, where $\underline{E}_{M_1} = [a_{ij}]$ and $\underline{E}_{M_2} = [b_{ij}]$. These are the entries of the i^{th} row of $\underline{E}_M$. Thus, $\underline{E}_M = \underline{E}_{M_1} \otimes \underline{E}_{M_2}$.

Now, assume that the theorem holds for $k \leq v-1 \geq 2$. For $k=v$, define the machine $M' = M_1 \times M_2 \times \dots \times M_{v-1}$. The theorem holds for M' , and thus $\underline{E}_{M'} = \underline{E}_{M_1} \otimes \dots \otimes \underline{E}_{M_{v-1}}$. But $M = M' \times M_v$ and the theorem holds for $k=2$. Hence, $\underline{E}_M = \underline{E}_{M'} \otimes \underline{E}_{M_v} = \underline{E}_{M_1} \otimes \underline{E}_{M_2} \otimes \dots \otimes \underline{E}_{M_v}$.

The following theorem follows from Theorems 1 and 6:

Theorem 7: A machine M can be embedded in a machine $R = R_1 \times R_2 \times \dots \times R_k$ if and only if there exists a coding matrix $\underline{A}$ such that

$$(\underline{E}_{R_1} \otimes \underline{E}_{R_2} \otimes \dots \otimes \underline{E}_{R_k}) \underline{A} - \underline{A} \underline{E}_M \geq \underline{0}.$$

Definition 9: By the machine R_j^k we mean the machine

$$R_j^k = R_j \times R_j \times \dots \times R_j, \quad (k \text{ factors})$$

where R_j is a shift register (one-way or two-way) of length j .

Definition 10: A set system on a set S is a collection of subsets (or called blocks) $\{B_1, B_2, \dots, B_k\}$ of S such that each element of S is in at least one subset of Q .

If a machine M can be embedded in a machine $R = R_j^1 \times R_1^k$ by the coding matrix $\underline{A}$, then $\underline{A}$ defines a set system on the state set S of M as follows: s_i is in block B_j if and only if there exists an integer r such that $(j-1)2^{k-1} \leq r \leq j2^{k-1}$, and $a_{ri} = 1$.

Definition 11: The product of two set systems Q_1 and Q_2 , denoted by $Q_1 \cdot Q_2$, is the set system consisting of the blocks which are the intersections of all possible pairs of blocks, one in Q_1 and the other in Q_2 .

Definition 12: A null set system, $\emptyset$, is a set system with exactly one state in each block.

Now, we can formulate an algorithm for multiple shift register realizations of sequential machines by finding all set systems (submachines) which are single shift-register realizable. Let M be the given machine with n states. The algorithm is given as follows:

- 1) Check if M is single shift-register realizable. If it is, STOP. If not, go to Step 2).
- 2) Set a limit $b = \lceil \log_2 n \rceil + 1$, where i is the maximum number of extra bits in any shift register in the realization over $\lceil \log_2(\text{number of blocks in the set system realized by the shift register}) \rceil$. Let $k=1$, $j=b-1$. Let V be a store of set systems with their codings. Set V empty. Go to Step 3).
- 3) Try to embed M in $R_j^1 \times R_1^k$. If successful, go to Step 4). Otherwise, go to Step 5).
- 4) Store the set systems defined by the codings generated by Step 3) in V , and go to Step 5). If any of these set systems have already been stored in V , discard the previous coding for the set system and go to Step 5).
- 5) Increase k by 1 and decrease j by 1. If $j=0$, go to Step 6). Otherwise, go to Step 3).
- 6) Pick a minimal set of set systems, $Q_1, \dots, Q_r$, from V such that $Q_1 \cdot Q_2 \cdot \dots \cdot Q_r = \emptyset$. The concatenation of the codings of $Q_1, Q_2, \dots, Q_r$ is a state assignment for the minimal shift-register realization.

V. CONCLUSIONS

In this paper, we have presented a general method for realizing synchronous sequential machines using a minimum number of shift registers.

The shift registers may be two-way and/or one-way, and the state assignment method is the most general type — many-to-one coding. The technique has been programmed, and it is very efficient for single shift-register realizable sequential machines. Further research is needed to increase the efficiency of the multiple shift-register realization method.

REFERENCES

- [1] C. L. Liu, "Sequential machine realization using feedback shift-registers", Proc. 5th Annual Symp. on Switching Circuit Theory and Logical Design, pp. 209-227; October, 1964.
- [2] A. J. Nichols, "Minimal shift register realization of sequential machines", IEEE Trans. on Electronic Computers, Vol. EC-14, No. 5, pp. 688-700; October, 1965.
- [3] W. A. Davis, "On shift register realizations for sequential machines", 1965 IEEE Conference Record on Switching Circuit Theory and Logical Design, pp. 71-83; October, 1965.
- [4] D. R. Haring, "On shift-register realizations of sequential machines and finite-state universal sequential machines", IEEE Trans. on Computers, Vol. C-17, No. 4, pp. 309-312; April, 1968.
- [5] C. C. Su and S. S. Yau, "Unitary shift-register realizations of sequential machines", IEEE Trans. on Computers, Vol. C-17, No. 4, pp. 312-324; April, 1968.
- [6] W. A. Davis, "Single shift-register realizations for sequential machines", IEEE Trans. on Computers, Vol. C-17, No. 5, pp. 421-431; May, 1968.

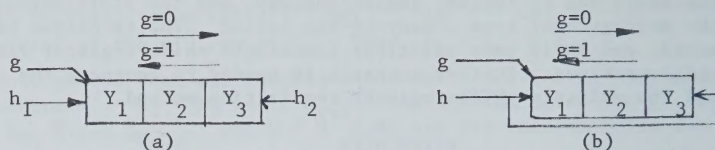


Fig. 1. Two equivalent schematic diagrams of a 3-bit two-way shift register.

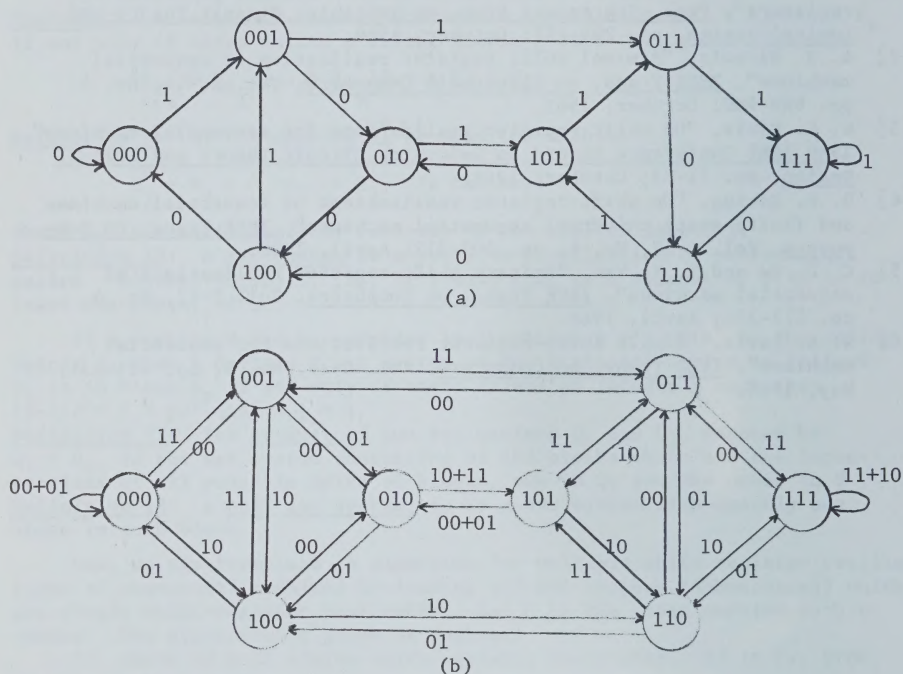


Fig. 2. The state graphs of (a) a 3-bit one-way shift register and (b) a 3-bit two-way shift register.

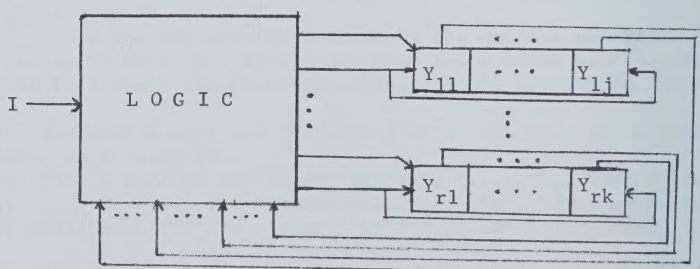


Fig. 3. The structure of a multiple two-way shift-register realization of a sequential machine.

i_1	i_2	i_3	$x_1x_2 \rightarrow 00 = i_1$	$01 = i_2$	$11 = i_3$		
s_1	s_2	s_1	-	$s_1 \rightarrow 0000$	1000	0000/0001	-
s_2	s_3	s_4	s_1	$s_1 \rightarrow 0001$	1000	0000/0010	-
s_3	s_7	s_6	s_2	$s_1 \rightarrow 0010$	1001	0001	-
s_4	s_1	s_5	s_2	$s_9 \rightarrow 0011$	1001	0001	0111
s_5	s_8	s_{11}	s_4	$s_4 \rightarrow 0100$	0010	1010	1000/1001
s_6	s_6	s_{10}	-	$s_{11} \rightarrow 0101$	0010	1010	1011
s_7	s_{12}	s_3	s_8	$s_7 \rightarrow 0110$	1011	1100	1101
s_8	s_7	s_5	s_{12}	$s_{10} \rightarrow 0111$	1011	0011	1111/1110
s_9	s_2	s_1	s_{10}	$s_2 \rightarrow 1000$	1100	0100	0000/0001
s_{10}	s_{12}	s_9	s_6	$s_2 \rightarrow 1001$	1100	0100	0010
s_{11}	s_1	s_5	s_{12}	$s_5 \rightarrow 1010$	1101	0101	0100
s_{12}	s_8	s_{11}	s_{10}	$s_{12} \rightarrow 1011$	1101	0101	0111
				$s_3 \rightarrow 1100$	0110	1110	1000/1001
				$s_8 \rightarrow 1101$	0110	1010	1011
				$s_6 \rightarrow 1110$	1111	0111	-
				$s_6 \rightarrow 1111$	1111/1110	0111	-

Figure 4. The state transition table of machine M_1 .

(a)

$x_1y_1y_2$	$x_2y_3y_4$
000 001 011 010 100 101 111 110	000 001 011 010 100 101 111 110
000	000
001	001
011	011
010	010
100	100
101	101
111	111
110	110

Figure 5(a) A coded state transition table of M_1 , and (b) the corresponding Karnaugh maps for h and g.

JOHN P. HAYES
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois

ABSTRACT

The problem of checking the equivalence of regular expressions is discussed and some existing approaches to this problem are examined. An equivalence-testing algorithm based on the construction of regular equations is described. It is shown to be particularly suited to a class of very long regular expressions.

INTRODUCTION

The behavior of a sequential machine can be described in a concise algebraic form by means of regular expressions, which were introduced by Kleene [1] in 1956. Up to the present, the practical value of regular expressions has been very limited. This is due in part to the fact that the same machine may be represented by an infinite number of distinct but equivalent expressions. Operations with regular expressions tend to produce very long complex expressions which are equivalent to much shorter ones. The problem of testing for equivalence is therefore of some interest.

The general approach to the equivalence problem is based on examination of $R \oplus S$, where R and S are the regular expressions being compared and $\oplus$ denotes exclusive-or. If it can be shown that $R \oplus S = \emptyset$ where $\emptyset$ is the empty set, then $R = S$, otherwise $R \neq S$.

An equivalence-checking algorithm has been described by Ginzburg [2]. His method involves using transition graphs, tables and regular equations and is therefore quite cumbersome. Furthermore it is limited to "restricted" regular expressions, i.e. expressions involving only the operators concatenation (denoted by $\cdot$ or juxtaposition), iteration (*) and union ($+$). General or "extended" regular expressions can contain any Boolean operator.

Another approach suggested (but not developed in detail) by Brzozowski [3] requires direct computation of all the characteristic derivatives [4] of $R \oplus S$. This procedure is perfectly general. However it tends to be unwieldy with long regular expressions since a derivative may be more complex than the expression from which it was obtained. Also determination of the similarity property [4] involves character-by-

character matching of derivatives with reordering in all possible ways of terms connected by +.

The algorithm described in this paper is based on the construction of regular equations. $R \oplus S$ is systematically decomposed into smaller expressions to yield a set of "transition" or T-equations. The T-equations are then combined to form the derivative equations of $R \oplus S$. It is proved in [5] and [6] that every regular expression can be characterized by a finite set of such equations. Equivalence is checked by inspecting these derivative or D-equations. The algorithm is general, i.e., is applicable to extended expressions, and seems more suited to the comparison of long regular expressions, both manually and using a computer, than the Brzozowski method.

OUTLINE OF THE ALGORITHM

Suppose R and S defined on the alphabet $Z = \{a_1, a_2, \dots, a_v\}$, are to be tested for equivalence. First, $R \oplus S$ is decomposed into shorter expressions by the application of certain rules, the particular rule applied depending on the form of $R \oplus S$. Variable names (X_i is used in the general case and $A, B, C, \dots$ are used in examples) are introduced to denote sub-expressions. Each variable defines a new equation which is in turn decomposed. Eventually a set of left-linear regular equations, i.e., equations consisting only of a union of left-linear terms, is obtained. (A left-linear term is an expression of the form λ where λ is the empty string, or $a_j X_i$ where $a_j \in Z$ or $a_j = \lambda$ and X_i is a variable name.) These left-linear equations are the T-equations.

Next, the T-equations are combined to form the D-equations of $R \oplus S$. These have the general form

$$Y_i = \sum_{j=1}^v a_j D_{a_j}(Y_i) + \delta(Y_i) \quad (1)$$

where the Y_i variables are formed by combining X_i terms. If $\delta(Y_i) = \lambda$ in any such equation, then $R \neq S$, otherwise $R = S$.

EXAMPLE

Example 1: Before giving the details of the algorithm, a simple example will be considered. Let $Z = \{0, 1\}$ and $A = (01)^* \cap (0 + 1)$. The D-equations of A will now be derived.

First, let $A = B \cap C$, where $B = 01^*$ and $C = 0 + 1$. We can write $B = 0D$ where D denotes 1^* . Using Arden's well-known theorem, we can write $D = 1D + \lambda$. If we introduce the equation $E = \lambda$, we can write $C = 0E + 1E$. Finally the empty set ϕ (denoted here by Y) can be defined by the equation $Y = 0Y + 1Y$. Thus we have the set of equations

$$\begin{aligned} A &= B \cap C \\ B &= 0D + 1Y \\ C &= 0E + 1E \\ D &= 0Y + 1D + \lambda \\ E &= 0Y + 1Y + \lambda \\ Y &= 0Y + 1Y. \end{aligned} \quad (2)$$

If we consider $B \cap C$ to be a compound variable, then equations (2) are left-linear; they are in fact our T-equations.

We now proceed to construct the D-equations. Substitute the expressions for B and C in the equation for A . This gives

$$\begin{aligned} A &= (0D + 1Y) \cap (0E + 1E) \\ &= 0(D \cap E) + 1(Y \cap E) \end{aligned}$$

which introduces the new variables $(D \cap E)$ and $(Y \cap E)$. Similarly from the equations defining $(D \cap E)$ we can derive

$$\begin{aligned} (D \cap E) &= 0(Y \cap Y) + 1(D \cap Y) + (\lambda \cap \lambda) \\ &= 0(Y \cap Y) + 1(D \cap Y) + \lambda. \end{aligned}$$

Continuing in this way we get a complete set of D-equations.

$$\begin{aligned} A &= 0(D \cap E) + 1(Y \cap E) \\ (D \cap E) &= 0(Y \cap Y) + 1(D \cap Y) + \lambda \\ (Y \cap E) &= 0(Y \cap Y) + 1(Y \cap Y) \\ (Y \cap Y) &= 0(Y \cap Y) + 1(Y \cap Y) \\ (D \cap Y) &= 0(Y \cap Y) + 1(D \cap Y) \end{aligned} \quad (3)$$

It will be noted that some of the equations in (3) are redundant, in fact, $(Y \cap E) = (Y \cap Y) = (D \cap Y)$.

The general rules for obtaining the D-equations of any regular expression R will now be developed.

THE TRANSITION EQUATIONS

First, R is denoted by some symbol X_1 and the equation $X_1 = R$ is written. The set of decomposition rules given below is applied in sequence to R and to all the new expressions generated by the rules. The particular rule which must be applied depends on the structure of the expression.

Rule 1: R is already in left-linear form. In this case it is left unchanged.

Rule 2: R is of the form aR_t where $a \in Z$ and R_t is any non-empty regular expression. Replace R by aX_i where X_i is a new, i.e., previously unused, variable symbol, and introduce the equation

$$X_i = R_t. \quad (4)$$

The decomposition rules are then applied to (4).

Rule 3: R is of the form $R_1^*R_t$. If the equation $X_i = R_1^*R_t$ has already been defined, then using Arden's Theorem replace it by

$$X_i = R_1X_i + R_t. \quad (5)$$

If R has not already been denoted by some variable symbol, then it is replaced by the new symbol X_i and equation (5) is introduced. In either case, the decomposition rules must be applied to each term on the right-hand side of (5).

Rule 4: R is of the form $(R_1 + R_2 + \dots + R_n)R_t$. If R_t is not already denoted by some X_t , then it is now replaced by X_t and the equation

$$X_t = R_t \quad (6)$$

is introduced. Then R is replaced by $R_1X_t + R_2X_t + \dots + R_nX_t$ using the distributive laws. The rules must now be applied to each R_iX_t term and to (6).

Rule 5: If none of the foregoing rules is applicable, then R must have the form $F(R_1, R_2, \dots, R_n)R_t$ where F is some Boolean function (not simple union). If R_t is not already denoted by some variable name, then the equation $X_t = R_t$ is introduced. R is replaced by $F(X_1, X_2, \dots, X_n)X_t$ and the equations

$$X_i = R_i, \quad i = 1, 2, \dots, n$$

are introduced. The rules must be applied to all new equations.

It should be noted that in applying the above rules, implicit use is made of the trivial equalities: $R = R\lambda = \lambda R$ and $R\emptyset = \emptyset$. Thus R_t may be λ in any of the rules.

The decomposition rules yield a set of left-linear equations. However, with extended regular expressions, the resultant equations may contain compound variables involving Boolean operators and concatenation but not iteration, e.g., $(\sim A \cdot B) \oplus C$.

Rules 1-4 are sufficient to handle restricted regular expressions. In this case, the equations obtained correspond directly to a transition graph for the original expression, hence their name. In fact, these rules

constitute an efficient method for obtaining a transition graph from a regular expression.

THE DERIVATIVE EQUATIONS

The T-equations as derived in the previous section, have the following general form:

$$X_i = a_j X_k + a_j X_\ell + \dots + a_k X_n + X_p + \dots + X_s + \delta(X_i) \quad (7)$$

where each $a_i \in Z$, each X_j is a simple or compound variable name, and $\delta(X_i) = \lambda$ or ϕ . The D-equations are formed in two stages:

Stage 1: Elimination of all X_j terms, i.e., all terms not beginning with some $a_i \in Z$, from the right-hand side of (7). Terms of this type result from applying rules 3 and 5. The elimination is done by substitution from the one or more equations defining X_j .

In the case of simple variables, this substitution is quite straightforward. Each X_j is defined by a single equation. The right-hand side of this equation is substituted for each occurrence of X_j .

Elimination of compound X_j terms is more difficult. Generally, X_j is of the form $F(X_1, X_2, \dots, X_n)$ where each X_i , $i = 1, 2, \dots, n$ is defined by a separate equation. Substitution for X_j requires forming the function F with the right-hand sides of the n defining equations as arguments. To do this, we use the following theorems:

Theorem 1: If $X = f(X_1, X_2, \dots, X_n)$ where f is a Boolean function and if

$$X_i = \sum_{j=1}^v a_j X_j^i + \delta(X_i) \quad \text{for } i = 1, 2, \dots, n,$$

then

$$X = \sum_{j=1}^v a_j f(X_j^1, X_j^2, \dots, X_j^n) + \delta(f(X_1, X_2, \dots, X_n)). \quad (8)$$

Theorem 2: If $X = X_1 \cdot X_2$ and $X_1 = \sum_{j=1}^v a_j X_j^1 + \delta(X_1)$,

then

$$X = \sum_{j=1}^v a_j X_j^1 X_2 + \delta(X_1) X_2.$$

Theorem 2 is trivial. The proof of Theorem 1 depends on the fact that $a_i X_i^k \cap a_j X_j^m = \phi$ for all i, j, k, m provided $i \neq j$. The theorem is easily proved when f is $\cap$, $+$ and $\sim$ (complementation); these form a

complete set of Boolean operators. The theorem then follows by induction on the number of operators in f .

A simple way to evaluate $\delta(f(X_1, X_2, \dots, X_n)) = \delta f$ in (8) will now be shown. Clearly $\delta(f(X_1, X_2)) = f(\delta(X_1), \delta(X_2))$ when f is $+$ or $\cap$. If we interpret $\sim\lambda$ as ϕ (instead of $Z^* - \lambda$) and $\sim\phi$ as λ (instead of Z^*), then we have $\delta(\sim X_1) = \sim\delta(X_1)$. With this rule we can define an operator $f'(\lambda, \phi)$ from any Boolean operator f , e.g., Sheffer stroke, in such a way that $f'(\delta(X_1), \delta(X_2)) = \delta(f(X_1, X_2))$. It follows by induction that $\delta f = f'(\delta(X_1), \delta(X_2), \dots, \delta(X_n))$ where f' is f with the modified interpretation of certain operators described above. This expression is easily evaluated. This special interpretation for $\sim\delta(X_1)$ may also be considered a consequence of axioms A_{13} and A_{14} for extended regular expressions given in [6].

As an illustration, consider the equations

$$\begin{aligned} A &= \sim B \\ B &= 0B + 1B \end{aligned}$$

which might be produced by the algorithm. Obviously $B = \phi$ and $A = (0 + 1)^*$. From Theorem 1 we can write

$$(\sim B) = 0(\sim B) + 1(\sim B) + \delta(\sim B). \quad (9)$$

If we use our rule, $\delta(\sim B) = \sim\delta(B) = \sim\phi = \lambda$. Equation (9) now implies that $A = 0A + 1A + \lambda$ which correctly defines $A = (0 + 1)^*$.

Every compound variable is composed of Boolean functions and concatenations of simple variables. Theorems 1 and 2 make it possible to obtain a left-linear equation defining any compound variable; this equation is then used to replace all occurrences of that variable. The derivation of a T-equation defining a complex X_j can be quite tedious as the foregoing theorems must be used repeatedly.

If substitution causes a term to appear more than once on the right-hand side of an equation, then all occurrences but one of the repeated term are eliminated, since $R + R = R$. If X_i reappears on the right-hand side of equation (7) which defines X_i , then this repetition can also be removed without affecting the equality. This is not immediately obvious. This type of repetition arises from nested star events of the type $(R^*)^*$. Suppose $A = (R^*)^*$. On applying the decomposition rules we get

$$\begin{aligned} A &= B + \lambda \\ B &= RB + A. \end{aligned} \quad (10)$$

Substituting the second equation in the first, we get $A = RB + A + \lambda$. Cancellation of the second A results in

$$\begin{aligned} A &= RB + \lambda \\ B &= RB + \lambda. \end{aligned} \quad (11)$$

Clearly from (11), $A = R^*$. But $R^* = (R^*)^*$, hence (10) and (11) are equivalent so the cancellation of the repeated A is valid.

Stage 2: Grouping of terms with the same "coefficient," i.e., beginning with the same a_i .

The T-equations have been reduced to the form:

$$X_i = a_j X_k + a_j X_l + \dots + a_m X_p + a_m X_q + \delta(X_i). \quad (12)$$

In the equation for X_1 , all terms with the same coefficient are grouped together as follows:

$$X_1 = a_j \sum_i X_i + a_k \sum_i X_i + \dots + a_m \sum_i X_i + \delta(X_1). \quad (13)$$

For brevity we write

$$X_1 = Y_1 = a_j Y_j + a_k Y_k + \dots + a_m Y_m + \delta(Y_1).$$

A new equation is added for each distinct Y_i . This equation is formed from union of the left-linear expressions defined by the X_i terms making up Y_j . The process is continued until no new Y_j terms are generated.

If (13) contains no term with coefficient $a_i \in Z$, then add the term $a_i Y$ and introduce the equation

$$Y = a_1 Y + a_2 Y + \dots + a_v Y$$

which defines $Y = \emptyset$.

The equations are now in the desired form; each variable denotes a derivative of R as in equation (1).

FURTHER EXAMPLES

Three regular expressions will be considered in this section.

Example 2: $R = (01^*0(011^*0)^*00)^*(01^*0(011^*0)^*(011^* + 0) + 01^*)$.

Example 3: $S = (01^*0)^*01^*$.

Example 4: $T = (\sim(0^*) \cap 01^*)^*$.

Application of the decomposition rules 1-4 to R yields

$$\begin{aligned}
R &= 0A + 0B + 0C \\
A &= 0D + 1A \\
B &= 0H + 1B \\
C &= 1C + \lambda \\
D &= 0E + 0F \\
E &= 1G \\
F &= 0R \\
G &= 0D + 1G \\
H &= 0J + 0K + 0L \\
J &= 1M \\
K &= 1N \\
L &= \lambda \\
M &= 0H + 1M \\
N &= 1N + \lambda.
\end{aligned} \tag{14}$$

Terms with the same coefficient are grouped together to form the D-equations. We set $Y = \phi$. Variables are renamed for brevity, e.g., P denotes $(A + B + C)$.

$$\begin{aligned}
R &= 0P + 1Y \\
P &= 0Q + 1P + \lambda \\
Q &= 0V + 1Y \\
V &= 0R + 1W + \lambda \\
W &= 0Q + 1W + \lambda \\
Y &= 0Y + 1Y
\end{aligned} \tag{15}$$

In a similar manner the D-equations characterizing S can be quickly derived.

$$\begin{aligned}
S &= 0X + 1Y \\
X &= 0S + 1X + \lambda \\
Y &= 0Y + 1Y
\end{aligned} \tag{16}$$

We now test R and S for equivalence by constructing the D-equations for $R \oplus S$ directly from equations (15) and (16) using Theorem 1. For brevity, let $X_i X_j$ denote $X_i \oplus X_j$.

$$\begin{aligned}
R \oplus S &= RS = 0PX + 1YY \\
PX &= 0QS + 1PX \\
YY &= 0YY + 1YY \\
QS &= 0VX + 1YY \\
VX &= 0RS + 1WX \\
WX &= 0QS + 1WX
\end{aligned}$$

Since none of these equations contains λ , we can conclude that $R = S$.

We now consider the extended regular expression T. The decomposition rules immediately yield the T-equations

$$\begin{aligned}
T &= (\sim A \cap B) \cdot T + \lambda \\
A &= 0A + 1Y + \lambda \\
B &= 0C + 1Y \\
C &= 0Y + 1C + \lambda \\
Y &= 0Y + 1Y.
\end{aligned}$$

The D-equations are most conveniently formed in two steps. First, using Theorem 1 we construct a set of equations defining $(\sim A \cap B)$ which is denoted by AB.

$$\begin{aligned} AB &= OAC + 1YY \\ AC &= OAY + 1YC \\ YY &= OYY + 1YY \\ AY &= OAY + 1YY \\ YC &= OYY + 1YC + \lambda \end{aligned}$$

Theorem 2 can now be used to form $(\sim A \cap B) \cdot T$ (denoted ABT).

$$\begin{aligned} T &= ABT + \lambda = OACT + 1YYT + \lambda \\ ACT &= OAYT + 1YCT \\ YYT &= OYYT + 1YYT \\ AYT &= OAYT + 1YYT \\ YCT &= OYYT + 1YCT + T \\ &= O(ACT + YYT) + 1(YYT + YCT) + \lambda \\ (ACT + YYT) &= O(YYT + AYT) + 1(YYT + YCT) \\ (YYT + YCT) &= O(ACT + YYT) + 1(YYT + YCT) + \lambda \\ (YYT + AYT) &= O(YYT + AYT) + 1YYT \end{aligned} \tag{17}$$

We can now compare T with, for example, R. From (15) and (17) we can write

$$R \oplus T = O(P \oplus ACT) + 1(Y \oplus YYT) + (\phi \oplus \lambda).$$

Since $\phi \oplus \lambda = \lambda$, we can say at this point that $R \neq T$.

REMARKS

The algorithm just described is similar to that of Ginzburg [2] despite differences in notation. The T-equations define a transition graph, and Ginzburg's pair equations correspond to the D-equations of $R \oplus S$. However the procedure used by Ginzburg to obtain the transition graph is not clear, and his method is limited to restricted regular expressions.

The present algorithm seems particularly suited to comparing long restricted expressions. The direct computation of derivatives is very tedious in such cases. Consider the computation of $D_0 R$ in Example 2. Using the rules for obtaining derivatives given in [4] we get

$$\begin{aligned} D_0 R &= 1^* 0(011^* 0)^* 00(01^* 0(011^* 0)^* 00)^* (01^* 0(011^* 0)^* (011^* + 0) + 01^*) \\ &\quad + 1^* 0(011^* 0)^* (011^* + 0) + 1^* \end{aligned}$$

which is longer than R itself. Thus, computation of derivatives may lead to an increase in complexity. In contrast, the decomposition rules break the given expression into simpler components.

Another difficulty with the direct computation approach is the test for similarity of derivatives to ensure that a complete set of

characteristic derivatives has been obtained. The analogous operation in the present algorithm is the comparison of the Y_i variables to ensure that a complete set of D-equations has been generated. This is much simpler, both manually and using a computer, than the similarity test. In a computer implementation, each variable X_i could be represented by a word of the form (0...010...0) which consists of all zero bits except for a 1 in the i -th bit position. Then $Y_k = X_i + X_j$ can be represented by (0...010.010...0), a word with 1 in the i -th and j -th positions only. To determine if Y_k and Y_ℓ are the same, form the exclusive-or of the corresponding words and test if the result is all zeros.

Extended regular expressions which consist of a Boolean function of restricted expressions cause little increase in difficulty, cf. Example 1. A single application of Theorem 1 suffices. Certain expressions, e.g., those containing iterated or concatenated extended expressions, are more difficult; repeated application of Theorems 1 and 2 may be required as in Example 4. The reason is that these expressions give rise to compound variables of the form $X_1 X_2 \dots X_k$ and, as indicated in [5], the number of equations needed to characterize such concatenated variables may be very large. A short extended expression such as $((\sim((\sim 1^*)0)^* 1)^*)^*$ may be easier to handle using Brzozowski's approach, i.e., by computing its derivatives.

It should be remarked that the decomposition rules given here are not necessarily the most economical for a given regular expression. Visual inspection of the sub-expressions provides many short-cuts, "obvious" equivalences may be detected and used to reduce the number of equations produced. Such ad hoc rules make for economy in short calculations and are often used implicitly, e.g., in [7]. However, with longer and more complicated expressions "direct" inspection no longer suffices; a systematic search of all previously derived sub-expressions must be instituted. This is of doubtful value, since in any case only simple equivalences, e.g., identities, could reasonably be detected. The choice is thus between generating a large number of simple equations with minimal searching, and generating a smaller number of more complicated equations with character-by-character matching.

It has already been observed that many of the equations generated by the algorithm define equivalent expressions and are therefore redundant. The D-equations of R directly specify a non-reduced flow-table of the finite-state machine which accepts R . Thus the detection and elimination of equivalent variables is essentially the state minimization problem. Any standard minimization technique may be used to eliminate

redundant equations; this can considerably reduce the length of an equivalence check.

ACKNOWLEDGMENT

The author would like to thank Professor Gernot Metze and Mr. Sharad Seth for their helpful criticism of this work.

REFERENCES

1. Kleene, S. C., "Representation of Events in Nerve Nets and Finite Automata," Automata Studies, Princeton Univ. Press, 1956, 3-41.
2. Ginzburg, A., "A Procedure for Checking the Equality of Regular Expressions," J. ACM 15 (April 1967), 355-362.
3. Brzozowski, J. A., Review of Ginzburg [2] in Comp. Rev. 8 (Sept.-Oct. 1967), 474.
4. Brzozowski, J. A., "Derivatives of Regular Expressions," J. ACM 11 (Jan. 1964), 481-494.
5. Salomaa, A., "Two Complete Axiom Systems for the Algebra of Regular Events," J. ACM 13 (Jan. 1966), 158-169.
6. Salomaa, A. and V. Tixier, "Two Complete Axiom Systems for the Extended Language of Regular Expressions," IEEE Trans. Computers C-17 (July 1968), 700-701.
7. Spivak, M. A., "Algorithm for Abstract Synthesis of Automata for an Expanded Language of Regular Expressions," Eng. Cybernetics 1 (Jan.-Feb. 1965), 49-59 (from the Russian).

K. C. WANG

Department of Electrical Engineering
Marquette University
Milwaukee, Wisconsin

ABSTRACT

This paper presents a unified approach to the synthesis of linear sequential circuits. The basic tools employed here are the coset partitions which are studied in detail and used to clarify the various synthesis methods proposed by many authors. They are then combined with the transition graphs to form a set of general synthesis methods which can be applied to test the linearity of any deterministic and completely specified flow table.

1. INTRODUCTION

Several approaches to the synthesis of linear sequential machines have been presented. Huffman (1) considered the transfer functions which are applied later by Hartmanis (2) to find the state assignments for linear machines having only two inputs. Other approaches may be classified into two categories: the transition graph approach, and the partition approach. The former methods include those of Elspas (3), Crowell (4), Gill (5,6), Hotz (7), Toda (8), LaVallee (9), and Wang (10) for autonomous linear machines, and those of Srinivasan (11), Yau and Wang (12) for non-autonomous linear machines. The latter methods include those of Cohn and Even (13), Brzozowski and Davis (14,15), and Hartmanis (2). Among these methods, it can be said that the synthesis of autonomous linear machines based on their transition graphs is very well acceptable. A set of elegant methods for synthesizing linear machines with coded outputs has also been established in (13) and (14). But none of them works satisfactorily if the machines are nonautonomous and with unspecified outputs. In order to make the synthesis techniques complete, naturally, more work needs to be done in this area. In addition to this, it is a general feeling that a great amount of similarities exists among the various methods using partitions. A unified treatment on this subject has been urgent. It is to these two objectives that the present paper is written. The basic tools to be employed in this study are the transition graphs and the coset partitions of linear machines. Since the former has been studied extensively

This work was supported by the National Science Foundation under Grant GK-1114.

elsewhere, only the latter will be considered here and used to clarify the various methods using partitions. A unified synthesis method using both the transition graphs and the coset partitions is then presented.

II. PRELIMINARY-- LINEAR MACHINES

A synchronous, deterministic sequential machine M is said to be linear if its inputs, states, and outputs can be identified with the elements of vector spaces (over the field $GF(2)$) and its state and output behavior described by a pair of matrix equations

$$s(t+1) = As(t) + Bx(t) \quad (1)$$

$$w(t) = Cs(t) + Dx(t) \quad (2)$$

The dimension of M , which will be denoted by n throughout this paper, is that of its state space S . Since this paper considers only the machines with unspecified outputs, (2) can be neglected, and we shall write (1) in the following convenient form

$$s' = As + Bx \quad (3)$$

for all s in S and x in the input space X .

The transitions of M are assumed to be specified by a flow table T in which each column corresponds to an input and each row to a present state. The entries of T are the next states under the corresponding inputs. Let the columns of T be denoted by M_k , $k=0,1,\dots,2^m-1$. If M is linear, then the transitions of each M_k are described by an affine transformation (16)

$$s' = As + s_{0k} \quad (4)$$

on S , where $s_{0k} = Bx_k$. Conversely, if there exists a state assignment such that each M_k is characterized by (4), then the transition part of M is linear. To code the inputs, we can choose a maximal set of independent s_{0k} and let the corresponding inputs form a basis of X . For a linear M , it is well known that any recoding of the states which corresponds to a nonsingular linear transformation preserves its linearity. An easy extension of this fact is that every nonsingular affine transformation on S also preserves its linearity. But the inputs may have to be recoded after the codes of S are changed. If S is changed by a nonsingular linear transformation, no change on the inputs is needed since the set of all the s_{0k} 's is still a subspace of S . If S is changed by a translation, the inputs can be recoded as follows: If s_t is a unit period state (11) under an input x_j , then the inputs are to be shifted by the translation $x \rightarrow x + x_j$ on X . If s_t is not a unit period state under any of the inputs, then the set of next states of s_t after the translation forms a coset of a

subspace. Therefore, we can add a component 1 to every input vector so that the dimension of X is increased by one and the new input vectors form a coset of the original X . In other words, we can regard T as being obtained from a flow table T having 2^{m+1} inputs by deleting the 2^m inputs that form a subspace of X . Similarly, we can also regard T as being obtained from a T having 2^{n+1} states by deleting the 2^n states that form a subspace of S . In this case the dimension of X remains unchanged but that of S is to be increased by one. Since the resulting state vectors form a coset of the original S , their components must span the constant row vector $a_c = (11 \dots 11)$. Thus, in synthesizing a linear machine, we may identify any input as the null input. Similarly for the null state. Although an incorrect choice of the null input leads to a realization of higher dimension, but the resulting state assignment must span a_c . By a theorem (Theorem 9) in (13), the dimension of S can be reduced by one at the expense of introducing a constant input. This result will be used later in our synthesis part.

III. COSET PARTITIONS

Definition 1. Let M be a linear machine. A coset partition (cp) on S is a partition P which divides S into coset blocks.

Coset partitions of linear machines were first studied in (11). They were also used implicitly in (2), (13-15) to find the linear state assignments. Our emphasis here will be placed upon their algebraic properties, their connections with the state assignment problem, and how are they being generated and used in the various synthesis methods using partitions.

Since S is a vector space, any cp on S can be thought of as being induced by a linear, or an affine, transformation. It is well known that the blocks of a cp are uniquely determined by its subspace block. Therefore, the cps induced by linear transformations having the same null space are identical. Similarly, the cp induced by an affine transformation is identical to that of its linear part.

Consider a linear machine M with unspecified outputs. The basic cps of M can be found from its flow table T as follows:

3a. The inverse images of the states in every M_k define a cp on S . Being induced by the equations $s' = As + s_{0k}$, these cps are identical. When M is nonsingular, they are the trivial cp ϕ in which every state is a block.

3b. Let x_0 be the null input with the corresponding column M_0 . Then the rows of M_0 and those of any M_k define a cp (M_0, M_k) which is on S if M

is nonsingular. Since any input can be treated as x_0 , it follows that the rows of any two columns form a cp (M_j, M_k) .

The set of cps is a closed set under the following operations:

1. Sum and intersection. Let $P_1 = (a_0, a_1, \dots, a_u)$ and $P_2 = (b_0, b_1, \dots, b_v)$ be two cps. Then $P_1 P_2$ is a cp; each of its blocks is the intersection of those of P_1 and P_2 , and $P_1 + P_2$ is a cp; each of its blocks is the union of those of P_1 and P_2 that are chain-connected.

2. Predecessors. Let P be a cp. Then the inverse image of P under M is a cp $\pi(P)$. If M is nonsingular, the predecessors of P form a cycle.

3. Cp sequence. Let P_1 be a cp. Then the members of the sequence $P_1, P_2, \dots, P_r$ are all cps, where $P_{i+1} = P_i + \pi(P_i)$ and $P_{r+1} = P_r$ is called the final cp of the sequence.

Theorem. Let M be nonsingular and consider the cp sequence defined above. Then P_r has the substitution property (S.P.), and is equal to the sum of P_1 and all of its predecessors.

Following the theorem, we see that $P_r = S$ if and only if the subspace block of P_1 contains a cyclic vector of A .

Corollary 1. Let $P_1 = (M_0, M_k)$ be a basic cp on S of a nonsingular M . Then in the sequence based on P_1 , $r=n$ and P_{n-1} is a two-block cp having no S.P. if and only if s_{0k} is a cyclic vector of A .

Corollary 2. Let $P_1 = (M_0, M_k)$ be a basic cp on S of a nonsingular M . Assuming that s_{0k} is a cyclic vector of the restriction of A on an invariant subspace S_r , then S_r constitutes the subspace block of P_r and its dimension r is equal to the least number of consecutive members in the cycle of P_1 needed to produce P_r .

We now present a unified treatment on the synthesis methods of (2), (13), (14) and (15), in terms of the cps. The basic principle is to find a set of coded cps which contains n independent row vectors not spanning a_c . When the outputs of M are coded, the desired cps are those induced by the outputs and, hence, coded directly by the outputs, and their predecessors. When the outputs are uncoded, it is then required to generate a set of codable cps, i.e., cps having two-blocks, and the methods become either too restrictive or too complicated.

IV. UNIFIED SYNTHESIS METHOD

In the synthesis part we consider the following problem: Given a

fully specified, reduced flow table T consisting of $2^{n-1} < S^{\#} \leq 2^n$ states, find a state assignment, if it exists, such that T is realizable as an n -dim. linear machine M . We also assume that T has more than one column and meets the necessary conditions for being linear given in (12) and (15).

We start with a brief review on the method of (12) which utilizes transition graphs. Assume that T has 2^n states and 2^m inputs ($m \leq n$) and hence, a unit period state s_0 . Identify s_0 as the null state and the corresponding input as the null input x_0 , which are placed in the first row and column respectively. Synthesize M_0 as an ALM according to its graph G and draw its coded graph G_c . Then identify the states in G with the codes in G_c subject to the transitions and the spanning relations specified by the cps (M_0, M_k) , for all k . This method is independent of the nature of M and also very efficient if applicable. But the restrictions imposed on S and X have considerably limited its usefulness. We shall show in the following that, with the help of the cps, all of the restrictions can be removed, and a set of general synthesis methods be established. We consider the following three cases separately.

Case 1. Nonsingular M

First we observe that the number of inputs need not be 2^m , provided that it does not exceed the number of states (12, 15). If the number of states is not 2^n , but T contains a unit period state, then the method of (12) is still applicable. In this case the graph G of M_0 is an incompletely specified ALM and we have to find a matrix A of order n such that its graph contains G . The states missing from T must form a closed set under the transitions and the additions by s_{0k} , for all k . Therefore, they do not affect the state identification process. However, if T contains no unit period state, then the method of (12) fails since we do not know the spanning relations among the states. In this case our synthesis steps are as follows:

Step 1. Identify arbitrarily an input as the null input x_0 . The graph G of M_0 is an incompletely specified ALM. Find a matrix A of order n , if $2^{n-1} < S^{\#} < 2^n$, or $(n+1)$, if $S^{\#} = 2^n$, such that its graph G_c contains G . Although this is a simple step, but as pointed out in (3), the matrix A may not be unique. If this is the case, we have to try each possible A in the subsequent steps. When $S^{\#} = 2^n$ but T contains no unit period state, the states appearing in T must form a coset of an n -dim. subspace. The dimension of the resulting assignment in this case is $n+1$, but the row vectors span a_c .

Step 2. Compute the $cp P_r$ by taking the sum of the basic $cp s_{jk} = (M_j, M_k)$ and all of their predecessors. Its subspace block S_r is then the least dimensional A -invariant subspace containing all the s_{0k} 's. Furthermore, P_r is also the final cp in the sequence of some basic cp , say, P_{01} . Then the code of s_{01} is a cyclic vector of the restriction of A on S_r . It is noted that none of the s_{0k} 's appears on T , but this does not affect our step. From P_{01} and its predecessors we can determine the cycle length of s_{01} . By Corollary 2, we can determine the value of r , the dimension of S_r . The code of s_{01} can now be found in G_c because its cycle contains exactly r consecutive independent vectors and has the length l , the cycle length of s_{01} . In case G_c contains more than one such cycle, we may identify s_{01} with any vector in these cycles; different choices are related simply by a changing basis of S_r .

Step 3. Identify the states in G with the codes in G_c subject to the transitions and the spanning relations specified by P_{01} . The state identification process is the same as that presented in (12). Repeat this step for each possible A , if necessary, until we have a valid assignment or conclude that M is not linear. For the case where the dimension of the assignment is $n+1$, we eliminate a row vector which is redundant in distinguishing the states to result in an n -dim. assignment.

Step 4. Compute the code of the translation state s_{0k} by the basic $cp P_{0k}$ and write down the transition equation of M_k . The inputs can be easily coded as described before.

Example 1. Consider the machine M_a shown in Fig. 1.

S	x_0	x_1	x_2
1	2	11	5
2	3	9	6
3	1	7	4
4	5	8	2
5	6	12	3
6	4	10	1
7	8	5	11
8	9	3	12
9	10	4	7
10	11	2	8
11	12	6	9
12	7	1	10

Fig. 1. Machine M_a .

It is seen that M_a is nonsingular and has no unit period state. We now synthesize it according to the steps just described.

1. Identify x_0 as the null input. The graph of M_0 is the cycle set

$G=(3,3,6)$ whose completed form must be $(1,1,2,3,3,6)=(1,1,2) \cdot (1,3)$, where "." stands for the cycle product (3). The matrix A is unique and given by

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Its graph G_c is the following cycle set:

$$G_c = \left\{ (0000); (1100); (1000, 0100); (0010, 0001, 0011); (1110, 1101, 1111); (1010, 0101, 1011, 0110, 1001, 0111) \right\}.$$

2. It is easy to show that the cp sequence based on $P_1 = P_{01} = (\overline{s_0, s_{01}}; \overline{2, 11}; \overline{3, 9}; \overline{1, 7}; \overline{5, 8}; \overline{4, 10}; \overline{6, 12})$ has $l=2$, and the final cp is $P_r = P_1 + \pi(P_1) = (\overline{1, 4, 7, 10}; \overline{2, 5, 8, 11}; \overline{3, 6, 9, 12})$.

Therefore, the code of s_{01} is in a cycle of length 2 and containing 2 independent vectors. We now look at G_c . The cycle of s_{01} is obviously $(1000, 0100)$. Suppose we choose $s_{01}=1000$.

3. Identifying s_7 as 1010, we have $s_8=0101$, $s_9=1011$, $s_{10}=0110$, $s_{11}=1001$ and $s_{12}=0111$. Then, using the relations $s_1 + s_7 = s_4 + s_{10} = s_{01}$, we have $s_1=0010$, $s_2=0001$, $s_3=0011$, $s_4=1110$, $s_5=1101$ and $s_6=1111$. Since the codes satisfy both the transitions and P_{01} , they form a valid assignment for M_a . The columns M_0 , M_1 and M_2 are described, respectively, by $s' = As$, $s' = As + (1000)^t$ and $s' = As + (0100)^t$.

Case 2. Nilpotent M

If T represents a nilpotent linear machine, then the graphs of its columns are identical trees. Because T is fully specified, each M_k must contain a unique unit period state which is the root of the tree. Therefore, the method of (12) is directly applicable to this case.

Case 3. Singular M

If T represents a singular linear machine, then the graphs of its columns are quasi-trees consisting of a set of identical trees planted on the cycle states. Consequently, T may not have any unit period state. In this case our synthesis method is to decompose T into its nonsingular and nilpotent components. The decomposition is done with the help of the cps as described in the following steps:

Step 1. Identify arbitrarily an input as x_0 . Construct the graph G of M_0 from which we form a partition P_c in the following manner: Let s be a cycle state and let $s_1, s_2, \dots, s_h$ be its successors in the cycle, where h is the height of the trees. Collect s and the i th level of the quasi-tree planted on s_1 , $i=1, 2, \dots, h$, as a block. Repeat this for every

cycle state. If M is linear, then P_c is unique and independent of the choice of x_0 . It was shown in (12) that P_c is a cp on S ; each of whose blocks contains a unique cycle state. Then construct the flow table T_c from T and P_c such that the states in T_c are the blocks of P_c . If we represent the blocks of P_c by the cycle states, T_c can be obtained from T by inspection. Since T_c represents the nonsingular part of T , it must be linearly realizable and we can apply the techniques described earlier to test its linearity. Assume that T_c is linear with the matrix A_c and a valid assignment distinguishing the blocks of P_c .

Step 2. Construct a partition P_t and a flow table T_t in the following manner: Collect all the cycle states as the block b_0 and let it be the null state in T_t . The images of b_0 in T are then disjoint sets of states which are also the blocks of P_t and the next states of b_0 . Similarly for their images. Continue this chain process until no new blocks can be generated. If this does not exhaust S , then collect a state, among the remaining ones, out of the same level of every quasi-tree to form the rest blocks. Enter these blocks and their next states into T_t . When M is linear, the resulting P_t is a cp on S ; each of whose blocks contains a unique state in the null tree t_0 . It is noted that t_0 may not appear in T at all but this does not affect our step. Now, T_t represents the nilpotent part of T , and we can apply the method of (12) to test its linearity. Assume that T_t is also linear with the matrix A_t and a valid assignment distinguishing the blocks of T_t .

Step 3. Since both P_c and P_t are coded cps and $P_c \cdot P_t = \phi$, their joint state assignment must be valid for the entire M . The overall matrix A is simply the direct sum of A_c and A_t .

VI. CONCLUSIONS

In this paper we have studied the coset partitions of linear machines and clarified the various synthesis methods using partitions. We have also extended the method of (12) to present a set of unified synthesis methods which are applicable to any kind of linear machines

REFERENCES

1. D. A. Huffman, "The synthesis of linear sequential coding networks," Information Theory, C. Cherry, Ed. Academic Press, Inc., New York, pp. 77-95; 1956
2. J. Hartmanis, "Two tests for the linearity of sequential machines," IEEE Trans. on Electronic Computers, Vol. EC-14, pp. 781-786; Dec., 1965.

3. B. Elspas, "The theory of autonomous linear sequential networks," IRE Trans. on Circuit Theory, Vol. CT-6, pp. 45-60; March, 1959.
4. R. H. Crowell, "Graphs of lineau transformations over finite fields," J. Soc. Indust. Appl. Math. Vol. 10, pp. 103-112; 1962.
5. A. Gill, "Analysis and synthesis of stable linear sequential cirucits," J. Assoc. Comput. Mach. Vol. 12, pp. 111-116; Jan., 1965.
6. A. Gill, "Graphs of affine transformations, with applications to sequential circuits." Conference Record of the 7th Annual Symp. on Switching and Automata Theory, pp. 127-135; Oct., 1966.
7. G. Hotz, "On the mathematical theory of linear sequential networks," Switching Theory in Space Technology, H. Aiken and W. F. Main, Eds. Stanford Univ. Press, Stanford, Calif., pp. 11-19; 1963.
8. I. Toda, "The tree set of linear machine," IEEE Trans. on Electronic Computers, Vol. EC-14, pp. 954-957; Dec., 1965.
9. P. Lavallee, "Nonstable cycle and level sets for linear sequential machines" IEEE Trans. on Electronic Computers, Vol. EC-14, pp. 957-959; Dec., 1965.
10. K. C. Wang, "Transition graphs of affine transformations on vector spaces over finits fields," J. Franklin Inst. Vol. 283, No. 1, pp. 55-72; Jan., 1967.
11. C. V. Srinivasan, "State diagram of linear sequential machines," J. Franklin Inst., Vol. 273, pp. 383-418; May, 1962.
12. S. S. Yau and K. C. Wang, "Linearity of sequential machines," IEEE Trans. on Electronic Computers, Vol. EC-15, pp. 337-354; June, 1966.
13. M. Cohn and S. Even, "Identification and Minimization of linear machines," IEEE Trans. on Electronic Computers, Vol. EC-14, pp. 367-375; June, 1965.
14. J. A. Brzozowski and W. A. Davis, "On the linearity of autonomous sequential machines," IEEE Trans. on Electronic Computers, Vol. EC-13, pp. 673-679; Dec., 1964.
15. W. A. Davis and J. A. Brzosowski, "On the linearity of sequential machines," IEEE Trans. on Electronic Computers, Vol. EC-15, pp. 21-29; Feb., 1966.
16. G. Birkoff and S. MacLane, A Survey of Modern Algebra, The Macmillan Co., New York; 1953.
17. J. Hartmanis and R. E. Stearns, Algebraic Structure Theory of Sequential Machines, Prentice-Hall, Inc., Englewood Cliffs, N. J.; 1966.

ON THE INDISTINGUISHABILITY OF FAULTS IN DIGITAL SYSTEMS

DONALD R. SCHERTZ and GERNOT A. METZE
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois

ABSTRACT

A discussion of the indistinguishability of digital faults as related to the detectability and locatability of these faults is presented. The relationship between the diagnosis of multiple and single faults is investigated, with emphasis on the determination of which multiple faults are indistinguishable from single faults.

INTRODUCTION

Various techniques have been proposed for the detection of faults in combinational networks.^{1,2} These techniques have typically included the generation of tests to detect all possible single faults and the formation of a covering table for the selection of an efficient set of tests. Such a table is then reduced by means of row and column dominance conditions and a final set of tests is selected.

Column dominance as applied to such a table is related to the relative detectability of faults. If two columns of the table are identical, then each may be said to dominate the other. The two faults corresponding to these columns are indistinguishable as any test which detects one will also detect the other. Thus, a bidirectional implication relationship may be said to exist between the detection of one fault and the detection of the other. If one column dominates another but is not equal to it, then the two faults corresponding to these columns are distinguishable. In this case, one fault is more readily detected than the other and the implication relationship is unidirectional.

The fault model considered here is a "stuck at zero-stuck at one" line model. In this model, each line in the circuit may be in one of three states: a) normal, b) stuck at zero (SAO), c) stuck at one (SAL). It is further assumed that a fault may occur at the input or output of any gate, on a primary input line, or on a primary output line. A shorthand notation will be used to represent faults. The fault "line 6 stuck at zero" will be written 6-0 and the fault "line 6 stuck at one" will be written 6-1.

FAULT COLLAPSING

The term fault collapsing will be applied to the process of combining faults by means of implication relationships derived from a study of the network concerned. Classes of indistinguishable faults are thus found before deriving diagnostic tests rather than after. Only one fault from each class need then be considered.

Fault collapsing will be considered in separate stages. Different stages will be used for different purposes; for example, the first and second stages are used when detection only is desired, but the first stage is used alone when the location of faults is necessary.

The first stage techniques of fault collapsing correspond to the case of identical columns in the covering table. In this case, the faults considered are indistinguishable and the implication relationships concerned are bidirectional. The second stage techniques correspond to the case of unequal columns. Here, the faults are distinguishable and the implication relationships concerned are unidirectional. The third stage techniques concern the relationships between single and multiple faults and therefore have no direct analogy in the single fault covering table problem.

Gate Collapsing

The various fault collapsing techniques will be discussed with the aid of the circuit of Figure 1. Consider the first stage analysis of the three-input AND gate shown. In order to detect the fault 4-0, the circuit inputs must be such that line 4 will be in the 1 state if the circuit is fault-free. For the AND this means that all inputs must be set to 1. Such a set of inputs will also detect the fault condition of any input stuck at 0. Conversely, any set of inputs which will detect an input line stuck at 0 will also detect the output line stuck at 0. The faults 1-0, 2-0, 3-0, and 4-0 are thus indistinguishable for this circuit. We write this symbolically:

$$1-0 \Leftrightarrow 2-0 \Leftrightarrow 3-0 \Leftrightarrow 4-0.$$

First stage fault collapsing as applied to a gate is sometimes included implicitly in a fault model as for example in the "output stuck at zero," "output stuck at one," "input open" fault model.

The second stage analysis of the same circuit deals with the remaining single faults for this gate. Consider the fault 1-1; in order to detect this fault, a test must place line 1 in the 0 state and all

other input lines in the 1 state. The normal output for this test will be a 0 and the test will thus also detect fault 4-1. The converse is not true as fault 4-1 might be detected by some test which places a 1 on line 1; such a test would not detect 1-1. A similar argument holds for any other input line stuck at 1. The second stage implication relationships for this gate are thus written:

$$1-1 \Rightarrow 4-1$$

$$2-1 \Rightarrow 4-1$$

$$3-1 \Rightarrow 4-1.$$

The third stage analysis of the AND circuit is concerned with multiple faults. The only multiple fault which leads to a useful implication relationship is the fault which consists of all input lines stuck at 1. This fault is indistinguishable from the single fault "output line stuck at 1," so we write:

$$4-1 \Leftrightarrow (1-1, 2-1, 3-1).$$

The foregoing analysis may be repeated for the OR, NAND, and NOR gates with "stuck at 0" and "stuck at 1" interchanged where appropriate. The case of the NOT gate is seen to be identical to that of a one-input NAND or a one-input NOR.

Fanout Collapsing

Fault collapsing techniques may also be applied at a fanout point in the circuit. First stage analysis is applicable only to the trivial case of a single line, that is, a line having a fanout of 1. In this case, we find that a fault at the origin of the line is indistinguishable from the corresponding fault at the destination of the line. This form of the fault collapsing is often included implicitly in a fault model.

Before attempting to apply second stage collapsing to a fanout point, we must classify the fanout. Reconvergent fanout describes a structure in which more than one path may be traced in a forward direction from one line to another in a circuit. That is, once a line has been split by fanout, the two new lines reconverge at some point further along in the circuit. Non-reconvergent fanout is defined in a similar manner.

In the case of non-reconvergent fanout, faults at the fanout point may be collapsed. Consider the fault 5-0 in the circuit segment of Figure 1. If line 5 does not reconverge with lines 6 or 7, then any test

which detects 5-0 will also detect 4-0. The converse is not true as fault 4-0 might be detected through line 6 or 7 rather than line 5. Thus, if lines 5, 6, and 7 do not reconverge, we write:

$$\begin{array}{ll} 5-0 \Rightarrow 4-0 & 5-1 \Rightarrow 4-1 \\ 6-0 \Rightarrow 4-0 & 6-1 \Rightarrow 4-1 \\ 7-0 \Rightarrow 4-0 & 7-1 \Rightarrow 4-1. \end{array}$$

The foregoing may not be true in the case of reconvergent fanout as the reconverging branch may interfere with the propagation of the fault condition to a primary output.

Third stage fault collapsing is also applicable at a fanout point in the circuit. Here we find that the multiple fault "all destination lines stuck at 0 (1)" is indistinguishable from the single fault "origin line stuck at 0 (1)." Thus the third stage relationships are written:

$$\begin{array}{l} 4-0 \Leftrightarrow (5-0, 6-0, 7-0) \\ 4-1 \Leftrightarrow (5-1, 6-1, 7-1). \end{array}$$

Application

The fault collapsing techniques presented are summarized in Table 1. In this table, each relationship is presented in a generalized form which does not use specific line numbers.

The first stage techniques presented above may always be applied whether the object of the analysis is detection or location. Transitivity may be used to combine classes formed by the implication relationships, and only one member of each class need be retained for test generation.

The second stage techniques are useful when detection only and not location is desired. That is, using a second stage relationship to eliminate a fault class will not cause the loss of information as to whether the circuit is good or bad, but may cause the loss of information as to which fault has occurred.

A word of caution is due here if the circuit concerned is redundant. In this case, it may happen that the dominated fault is indistinguishable from the good machine but the dominating fault could be detected. If the second stage technique is used to eliminate the dominating fault, no test will be found for this fault even though one exists.

The third stage techniques are useful in cases where multiple as well as single faults are to be considered. If all multiple faults are to be detected, these relationships may be used to reduce the number of single faults which form the basis for the expansion into multiple faults.

After applying as many stages of fault collapsing as desired, a set of classes of indistinguishable faults remains. One fault must be chosen to represent each class for the generation of tests. Various criteria may be useful in making this choice. For example, one might choose to minimize the number of "stuck at 1" or "stuck at 0" faults picked as class representatives. Another choice would lead to the minimization of the number of remaining fault locations. This choice is most desirable as it tends to make the test generation process simpler.

MULTIPLE FAULT DETECTION

The diagnosis of combinational circuits is usually carried out under the single fault assumption in order to take advantage of the drastic reduction in the number of fault combinations which have to be considered; from 3^p multiple faults on p lines to $2p+1$ single faults. The single fault assumption however cannot be justified in such applications as acceptance testing. It would therefore be of considerable interest to determine which multiple faults would be detected by a set of tests which was derived under the single fault assumption.

The problem just outlined may now be considered based on our indistinguishability classes of single faults. Before attacking the problem, we will look further into the characteristics of the indistinguishability classes derived at the first stage of fault collapsing.

The indistinguishability classes are formed by applying transitivity to expressions like those of the first stage of Table 1. The faults in each class form structures of the types shown in Figure 2a, 2b, 2c. These structures, which either narrow or remain a constant width as one progresses in a forward direction through the circuit, are all topologically equivalent to the form shown in Figure 2d, which we shall choose as a representative form for any indistinguishability class. Each class begins with one or more faults at the level nearest the input and ends with a single fault nearest the output of the network.

Neighboring indistinguishability classes can be situated such that the final fault in one class is at the same fault position as one of the initial faults of another class. In this case, we say that the two classes are strongly connected. An example of two strongly connected classes is shown in Figure 3a. If two classes are situated such that a chain of strongly connected classes exists between them, they are said to be connected. In Figure 3b, all classes are connected.

A final property of indistinguishability classes is concerned with path sensitization through the circuit. Consider the AND gate of Figure 1; the indistinguishability classes formed are 1-0,2-0,3-0,4-0; 1-1; 2-1; 3-1; 4-1. The first of these classes spans the gate, that is, involves faults on both the inputs and the output of the gate. (There will always be one and only one fault class which spans a gate.) For each input line in the class spanning the gate, we now define a sensitizing line value opposite the fault value listed for that line. In the example above, the sensitizing line value is 1 for each of the three input lines. In order to sensitize a path through one input line of a gate, every other input line must be set to its sensitizing line value. If we wish to sensitize a path via input line 1 in the gate of Figure 1, we must place the value 1 on lines 2 and 3.

The properties of connectedness and sensitization described above may now be utilized to consider multiple faults. In the following analysis, we will assume that we have a set of tests which detects all single faults in the circuit. Such a set can always be found if the circuit is non-redundant.

Multiple faults composed of single faults in the same indistinguishability class will be detected by any test which detects the single fault which is the class representative. This is readily seen by considering that such a test must place the line sensitizing value on every line in the class.

Next we consider two strongly connected classes as seen in Figure 3a. Any multiple fault consisting of two or more faults from the same class will be detected as above. Any multiple fault consisting of one or more faults from each class will also be detected as the fault or faults in the class closest to the output will effectively block the other faults and thus will be detected. Any single or multiple fault from two strongly connected classes will therefore be detected.

Finally, we consider a set of connected classes as in Figure 3b. Multiple faults falling in the same class or in strongly connected classes have previously been considered so we only need concern ourselves with faults falling into classes which are not strongly connected.

If the single faults composing a multiple fault fall into classes which are along the same path through the circuit, then an argument exactly analogous to that for the case of strongly connected classes above may be used to show that the multiple fault is detected by the test for the single fault nearest the output.

The only multiple faults remaining are those in classes which are not along the same path through the circuit. Consider, for example, a single fault in the first level of Figure 3b. This single fault is guaranteed to be detected by some test which will place specific sensitizing values on all lines in the circuit. Any other faults which do not change the value of any of these lines will not affect the response of the circuit and the original fault will be detected. If the additional fault or faults do change the value of some line, there are only two possibilities: the additional fault does not alter the sensitization pattern in the circuit, in which case the original fault is detected; or the additional fault does affect the sensitization pattern. The sensitization pattern can only be changed by an off-path fault by changing the output line to the fault value, in which case the new fault is detected. Thus we have shown that if all single faults are detected by a set of tests, then the same set of tests will also detect any multiple fault within any single connected set of indistinguishability classes.

A general representation of a circuit consists of one or more sets of connected indistinguishability classes which are separated because of fanout. However, since any multiple fault within a connected set will be detected, the third stage relationships may be used to prove that some multiple faults from unconnected sets will also be detected. In particular, in the two level realization of any single output function, the fault class structure takes the form of one large connected set of classes and a number of small sets. In this case, the third stage techniques may be utilized to reach every fault class in the small sets and thus it may be shown that every multiple fault will be detected by a set of tests which was derived to detect all single faults.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant NSF GK-1663. Mr. Schertz was further supported by a National Science Foundation Science Faculty Fellowship.

REFERENCES

1. J. P. Roth, W. G. Bouricius, and P. R. Schneider, "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits," IEEE Trans. on Electronic Computers, Vol. EC 16, No. 5, October, 1965, pp. 567-580.
2. J. F. Poage, "Derivation of Optimum Tests to Detect Faults in Combinational Circuits," Mathematical Theory of Automata, Polytechnic Press, Brooklyn, N.Y., pp. 483-528, 1963.

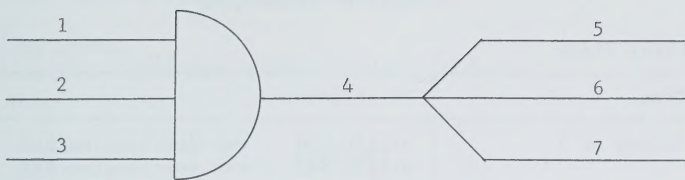
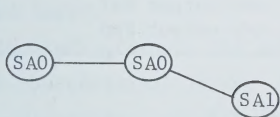
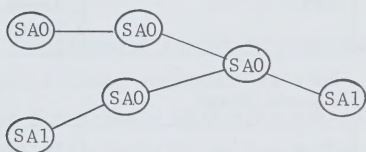


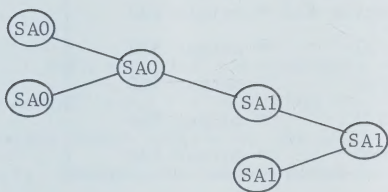
Figure 1



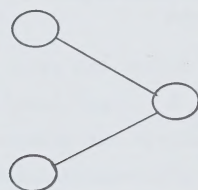
a.



b.

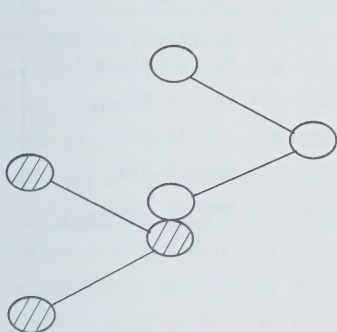


c.

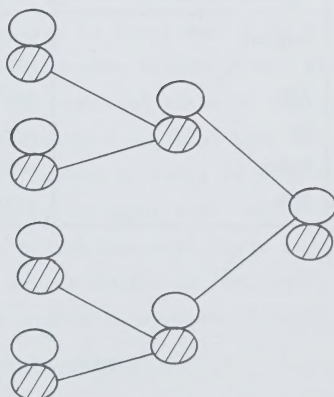


d.

Figure 2



a.



b.

Figure 3

SUMMARY OF TECHNIQUES

First Stage

Type	Description
fanout of 1	origin SA0 $\Leftrightarrow$ destination SA0 origin SA1 $\Leftrightarrow$ destination SA1
AND	any input SA0 $\Leftrightarrow$ output SA0
OR	any input SA1 $\Leftrightarrow$ output SA1
NAND	any input SA0 $\Leftrightarrow$ output SA1
NOR	any input SA1 $\Leftrightarrow$ output SA0
NOT	input SA0 $\Leftrightarrow$ output SA1 input SA1 $\Leftrightarrow$ output SA0

Second Stage

Type	Description
non-reconvergent fanout	any destination SA0 $\Rightarrow$ origin SA0 any destination SA1 $\Rightarrow$ origin SA1
AND	any input SA1 $\Rightarrow$ output SA1
OR	any input SA0 $\Rightarrow$ output SA0
NAND	any input SA1 $\Rightarrow$ output SA0
NOR	any input SA0 $\Rightarrow$ output SA1

Third Stage

Type	Description
fanout	origin SA0 $\Leftrightarrow$ all destinations SA0 origin SA1 $\Leftrightarrow$ all destinations SA1
AND	output SA1 $\Leftrightarrow$ all inputs SA1
OR	output SA0 $\Leftrightarrow$ all inputs SA0
NAND	output SA0 $\Leftrightarrow$ all inputs SA1
NOR	output SA1 $\Leftrightarrow$ all inputs SA0

Table 1

SYNTHESIS REQUIREMENTS FOR FAULT DETECTION

THEO JAY POWELL

Coordinated Science Laboratory, Urbana, Illinois

ABSTRACT

Diagnosis of combinational logic can be divided into analysis and synthesis of tests. Analysis applies a given input vector to the logical circuit and determines which faults can be detected by the input vector. On the other hand synthesis generates a test input that will detect a specified fault.

This paper presents the essential requirements needed to generate tests in combinational logic. Knowledge of these requirements leads to a better understanding of what is required for a fault detection technique and helps one develop a good process. Also included is an algorithm for detecting faults in combinational logic.

INTRODUCTION

Both analysis and synthesis of diagnostic tests are useful in forming a set of tests for detecting faults in combinational logic. A test is a vector input applied to the logic in order to examine and compare its output. Analysis is useful since it easily indicates the specific output produced by an input test applied to the logic with a fault. Several faults are detected by many tests, and these can be quickly separated by analysis of a few tests. However, sometimes certain faults are detected by only a few tests; only one test at times. Unless one is lucky and tries the right test, an analysis procedure may require considerable computational effort in analyzing individual tests before one of the required tests is processed. Synthesis is extremely useful since it generates a required test directly, and the need to systematically try a large number of tests is eliminated.

Here synthesis will be the generation of an input test vector that will detect a given fault if it occurs in a combinational logic circuit. Knowing the basic essentials required to detect faults in combinational logic should help to develop an efficient method to synthesize tests.

FAULT DETECTION REQUIREMENTS

Fault Model

The first essential is a proper fault model. The more common types of faults that can occur in combinational logic are shorted and opened transistors. Logically these are equivalent to an input or output line being

stuck at 1 or 0. Other types of faults exist, such as lines shorting together and intermittent faults. However, one must decide how close one wants to model the real world and what faults are the most prevalent. Of course, the closer the model is to actual circumstances the more complicated the model and the greater the complexity in using it. This paper will follow the majority of the papers and treat well-defined faults, i.e., not intermittent, and the stuck at 1 or 0 faults. Furthermore, in order to considerably simplify the problem we will assume that only one fault occurs at any time. This is a reasonable assumption if the logic is tested often enough so that the probability of two or more faults occurring is very small. However, many multiple faults will be detected by the tests chosen for single faults.

Circuit Knowledge

A knowledge of the logic interconnections is needed. Although it is doubtful that any person would do the following, it is possible to design an AND gate that requires every possible input combination to verify whether it is fault free. Fig. 1 shows an OR gate with two inputs which requires all four input vectors to detect all possible single faults.

Furthermore, it is possible to design a logic circuit that will require any possible input test to detect certain faults.

In Fig. 2 assume the output of NOT N_1 is stuck at 1. The detection of this fault requires $x_j = 1$, $f_t = 1$ where f_t is any function of the inputs, and $f_n = 0$ where f_n is the fault-free function. The regular function f_n will be blocked by N_2 and A_2 which will cause the output to be dependent on the fault. Since $f_n = 0$ the output for the fault-free circuit would be 0 while for N_1 stuck at 1 it would be 1. Detection of this fault requires $f_t = 1$ where f_t can be any function of the inputs for which $f_n = 0$. With the line from A_1 to O_1 removed, detection of this fault requires $f_t = 1$ where f_t is any function for which $f_n = 1$.

A designer would not use such design principles; however, interconnections of logical elements may inadvertently create networks which require fault-detection tests solely due to the interconnections. Therefore, unless the logic circuit is known, functional testing is required to be certain that all single faults are covered. Functional testing is the applying of all the possible input vectors that the circuit may use and checking them for verification with their fault-free outputs. Since we are trying to obtain a smaller set of detection tests than the number of functions used, a knowledge of the circuit's interconnections is needed.

Dependent Path

The next essential is paths from the inputs to the logical element being tested and at least one path from the tested element to an output of the circuit. It is possible to have a gate in the circuit that is not fed by a path from the primary inputs or does not have any path from the tested element to a primary output; however, such a gate would have no effect on the output and would therefore be meaningless. Armstrong's path sensitizing procedure [1] emphasizes the path approach and generates tests to detect faults in combinational logic without reconverging fanout problems. Reconverging fanout will be discussed later.

The primary inputs via the first paths must apply the required input configurations to the element being tested, called the tested element. The path from the tested element to the output must have a dependence on the element's output. If the output of the element changes then the circuit's output must change. This requires every AND or NAND element along the sensitized path to have gate inputs of 1 on all inputs not on the path. Every OR or NOR along the sensitized path must have gate inputs of 0 on the inputs not along the path. Of course, a NOT does not have any other inputs, and so its output is dependent on its one input.

Roth's algorithm [3,4,5] makes extensive use of the element dependence by chaining together elements whose outputs are dependent on given sensitized inputs and thereby forming a path to the output.

Reconverging Fanout

Reconverging fanout occurs when there exists more than one path from the tested element to the same output. It is caused by an element, along the sensitized path after the tested element, having multiple fanout and then another element recombining two or more paths before the output is reached.

While tests are being generated using the element dependence, we must not overlook two problems that can arise from reconverging fanout. Care must be taken in generating detecting tests since one path sometimes can block the dependence of the fault in another path. Other times both reconverging paths must be sensitized to detect a fault.

The problem of two or more paths blocking the output's dependence of the fault can occur if the difference in the number of negations between the sensitized path and the reconverging path from the tested element to the output is odd. At the point of reconvergence the function f_r would have the form:

$$f_r = x_f f_1 \vee \bar{x}_f f_2 \vee f_3$$

where x_f is the output of the tested element and f_1 , f_2 and f_3 are functions of the other inputs and not a function of x_f . For f_r to be dependent on the fault requires $f_3 = 0$. If an assignment of inputs causes $f_1 = f_2 = 1$ then $f_r = x_f \vee \bar{x}_f = 1$. The output would then not be dependent on the fault. To prevent this from happening requires either $f_1 = 0$ and $f_2 = 1$ or $f_2 = 0$ and $f_1 = 1$ as well as $f_3 = 0$. This output dependence requires that somewhere along one of the reconverging paths an element's output must not be dependent on the fault. If the path crosses an AND element whose output must be 0, an element input other than the fault-dependent input must be 0. A 0 will also block the dependence of a sensitized input for NAND and a 1 will block the dependence for an OR or NOR.

If neither of the two paths can be blocked while the other path is sensitized and there are no other paths, the machine containing the fault is indistinguishable from the fault-free machine.

Consider Fig. 3 for an example of this problem.

To detect faults associated with element e , assume the path $e g i$ for detection where $i = ed \vee \bar{e}a$. The sensitization of path $e g i$ requires $d = 1$ making $i = e \vee \bar{e}a$. If $a = 1$, then $i = 1$ and is not dependent on any fault associated with element e . But for $a = 0$ and $d = 1$, $i = e$, and any single fault occurring at e will be detected at i .

The other problem in reconverging fanout that one must be aware of occurs when two or more paths are required to detect faults. There is a danger that when two or more paths do exist one may try to sensitize each path by itself and fail. At this point one may be tempted to declare the fault undetectable, i.e., the machine containing this fault having the same truth table as the fault-free machine. However, one cannot safely call the fault undetectable until one has tried to sensitize multiple paths simultaneously. Note that detecting faults by multiple paths requires the difference between two reconverging paths to be even, otherwise the output dependence of the fault is cancelled.

An example of a circuit requiring multiple sensitized paths for fault detection is shown in Fig. 4.

To detect the fault "b to e open" in Fig. 4 requires $b = 0$, $c = 1$, $a = 1$, and $d = 1$. Element outputs g and f are both 0 when output e is 0, and both 1 if e is 1 because b is open. The output h is then 0 if the circuit is fault-free and 1 if b to e is open. The sensitized paths are both $e f h$ and $e g h$, and if one path is blocked, then the output h is no

longer dependent on the output of e.

Gate Input Vectors

To detect stuck at 1 or 0 conditions requires, for each logical gate, $n+1$ gate input vectors where n is the number of gate inputs. For AND (NAND) the logical gates must have a gate input vector of all 1's to verify that the output is not stuck at 0 (1) for the AND (NAND). This vector of all 1's activates the output. For each input line there must be an input vector that has a 0 on this line and 1's on all other lines. These n gate input vectors verify that none of the input lines are open and that the output is not stuck at 1 (0) for the AND (NAND). Verifying that the input lines are not open also verifies that the individual input lines are not stuck at 1. The fault of any one of the input lines stuck at 0 is equivalent to the output being stuck at 0 (1) for the AND (NAND), and there is no need to generate a separate test to check for this fault.

By duality, OR's (NOR's) require a gate input vector of all 0's to check that the output is not stuck at 1 (0) and, for every line, input vectors of a 1 on the given line with 0's on the other lines. The latter vectors verify that the input lines are not open or that each line will activate the output by itself.

Each of the mentioned vectors are required to detect a possible fault and any other gate input vectors will not detect single faults not detected by this set.

FAULT DETECTION PROCEDURE

These essential requirements for detection can be combined to produce an efficient procedure to generate fault detection tests for combinational logic. Briefly this procedure is as follows: First, formulate all the gate input vectors that are required to detect the faults at the gate's output and then one by one generate the required primary inputs to detect the possible faults at the primary outputs. For each gate input vector all possible paths from the logical element to a primary output are listed. Generation of the primary inputs that will sensitize the shortest path is first attempted. If the attempt is successful, the test and path is stored and the next gate input vector is considered. If the attempt is unsuccessful, then the next shortest path is chosen until all possible paths are exhausted. If the attempts on all paths are unsuccessful, the fault which the gate input vector detects is declared undetectable and the next required gate input vector is considered.

In generating the primary inputs, the values 1 or 0 on the individual lines are arbitrary for some logical gates and not for others. For

example, if an AND must have a 1 output, then all its inputs must have the value 1. If an AND which has n input lines must have a 0 output, then there are $2^n - 1$ possible choices of gate input vectors to choose from that will cause the output to be 0.

While assigning line values in the detection procedure, the definitely specified lines are assigned their values first. The arbitrarily specified lines are then assigned their values by choosing, as much as possible, input vectors that are the same as gate input vectors required to detect faults. While specifying line values from logical gate to logical gate along the path to the output, previously generated tests are compared to see if their sensitized paths and the sensitized path of the test being generated intersect. If they do, the specified line values are compared to see if they are the same. If not, the assigning of line values is continued; however, if the specified line values are the same, the previously generated test also detects the fault for which we are presently generating a test. Since the sensitized paths intersect, the output will change if either of the faults covered by the test occurs. This comparison reduces the overall number of tests required to detect the faults of the circuit, and since whenever possible gate test vectors are used, the probability of combining tests is increased.

After specifying the values on the lines to detect a given fault, a check must be made to verify that the output's dependence has not been nullified by a second path, i.e., the first of the reconverging fanout problems. A quick check to see if the second path could nullify the dependence is made by taking the difference in the number of negations of the two paths. If the difference is an even number, then nullification cannot occur; however, if the difference is odd, then the second path must be blocked by other inputs if possible. If the second path cannot be blocked, another path is chosen in an attempt to find a test that will detect the fault after trying all paths the fault is declared undetectable.

While generating values for input lines, apparent line value contradictions may arise. A line value contradiction occurs when one value (0 or 1) is required to extend the fault dependence through another logic gate but the opposite value has already been placed on the line due to other detection requirements.

Before abandoning a path because of an apparent line value contradiction, examination must be made to see if the existing value on the line has been caused by a multiple path dependence. If multiple path dependence does exist, then the value already existing on the line question need not be

changed. The logical gate's output at which the two dependent lines reconverge will change if the considered fault occurs because both dependent input lines will change. The test generating procedure then handles multiple paths by dealing with one of the paths. If multiple path dependence is required for detection, the other path will automatically be sensitized at the same time.

After detecting all gate input test vectors at the outputs, an efficient test set is obtained. Several of the tests cover the detection of more than one fault, and therefore the effort in reducing the set of detection tests to a still smaller set is lessened. There is no guarantee that the smaller set will be the smallest set that detects all the faults. The procedure will not always generate the tests that cover the most faults; however, the smaller set will be less than the number of single faults and for most purposes a further reduction would not be justified.

Example. Consider the circuit in Fig. 5. The gate input vector tests are listed in Fig. 6. The path for the first element g is g j m. Any fault to be detected at j requires a 0 on h, and to be detected at k requires 1's on k and i. A 1 on i requires e and f to have 0's. A 0 on h can be obtained by three possible gate input vectors on h. The test input configurations 5 and 6 are chosen and are alternated in use. See tests 1, 2 and 3 in Fig. 7.

Element h has two possible paths to detect its faults. If path h j m is chosen, the path dependence requires a 0 to be on g and 1's on k and i. However, the 1 on h requires k to be a 0 which is a line value contradiction. Since the difference in the number of negations between the two paths is odd, no test using both paths can be generated for this fault. Therefore the other path must be tried. The path h k m requires 1's to appear on i and j. Since h is a 0, a 1 on g is required. The remaining required values are shown in test 4b which detects the fault of h stuck at 0.

If path h j m is selected for tests 5 and 6, a 0 must appear on g and 1's on k and i. Possible values appear in tests 5a and 6a; however, note that if the fault h stuck at 1 appears, then k is a 0 and m remains 0 instead of changing to 1. The output is not dependent on h because the reconverging paths have an odd difference in their number of negations. The second path h k m must be tried. The value 1 must appear on i and j. The 1 on j requires g to be a 1 which also prevents j from being dependent on h. Therefore m is dependent on h and 5b and 6b detect the faults "c to h open" and "d to h open."

Test 7 has the path i and requires 1's on j and k . The paths in test 1 and test 7 intersect at m , and the line values already assigned to test 7 are the same as in test 1. Therefore test 1 covers test 7 and the next test is considered. The fault that gate test 11 detects is undetectable at the output since a 1 on h requires k to be 0 which prevents m 's dependence on j . The other tests are generated similar to the above examples.

SUMMARY

There is a role for synthesis as well as analysis in the detection of faults in combinational circuitry. Therefore, knowing the basic essential requirements for the generation of tests allows one to develop an input test set that usually has fewer tests than that required for functional testing.

Basic requirements for synthesis of fault detection are as follows: A realistic fault model and a knowledge of the logical interconnections. A path from at least one input to the logical element being tested and a path from the tested element to a primary output. Along this path a dependence on the fault must exist, or in other words, if the fault occurs, then the outputs along the path must all change. If reconverging fanout occurs, then the two problems associated with it must be dealt with. The first is that if two or more paths are sensitized by the same element being tested, then there cannot be an odd difference in the number of element negations between the paths. The other problem is that one must not overlook cases when detection requires two or more of the paths to be sensitized for output dependence.

Generating tests for stuck at 1 or 0 conditions for a gate with n inputs requires $n+1$ tests. The tests must apply to the gate inputs, vectors that will verify that each individual line will by itself inhibit or activate the output (depending on whether the element is an AND, NAND, OR, or NOR) and a vector that will verify that the output is not stuck in the inactive condition. The procedure described in the paper uses these essential properties while generating a small set of tests for testing combinational logic.

ACKNOWLEDGMENT

The author would like to express appreciation to Professor Gernot Metze for his suggestions and criticisms and to NASA for their financial support of this research.

REFERENCES

1. D. B. Armstrong, "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets," IEEE Trans. on Elec. Computers, Vol. EC-15, No. 1, pp. 66-73 (February 1966).
2. R. D. Eldred, "Test Routines Based on Symbolic Logical Statements," J. ACM, Vol. 6, pp. 33-36 (January 1959).
3. J. M. Galey, R. E. Norby, and J. P. Roth, "Techniques for the Diagnosis of Switching Circuit Failures," IEEE Trans. on Comm. and Electr., Vol. 83, No. 74, pp. 509-514 (September 1964).
4. J. P. Poage, "Derivation of Optimum Tests to Detect Faults in Combinational Circuits," Proc. Symposium on Math. Theory of Automata, pp. 483-528 (Polytechnic Press, New York, N.Y., April 1963).
5. J. P. Roth, "Diagnosis of Automat Failures: A Calculus and a Method," IBM J. Res. Dev., Vol. 10, No. 4, pp. 278-291 (July 1966).
6. J. P. Roth, W. G. Bouricius, and P. R. Schneider, "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits," IEEE Trans. on Computers, Vol. EC-16, No. 5, pp. 567-580 (October 1967).
7. S. Seshu and D. N. Freeman, "The Diagnosis of Asynchronous Sequential Switching Systems," IRE Trans. on Elec. Computers, Vol EC-11, pp. 459-465 (August 1962).
8. S. Seshu, "On an Improved Diagnosis Program," IEEE Trans. on Elec. Computers, Vol. EC-14, No. 1, pp. 78-79 (February 1965).

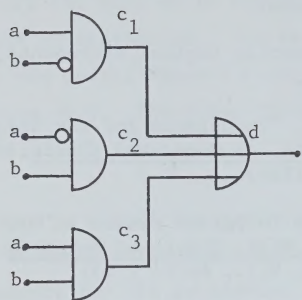


Fig. 1

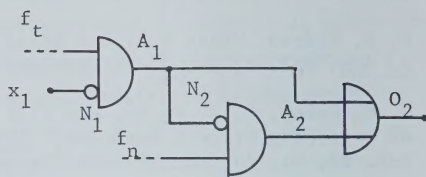


Fig. 2

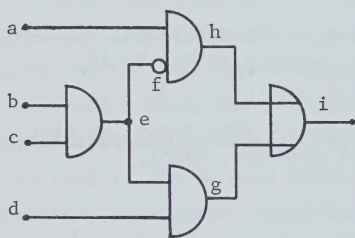


Fig. 3

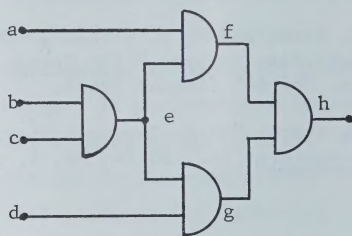


Fig. 4

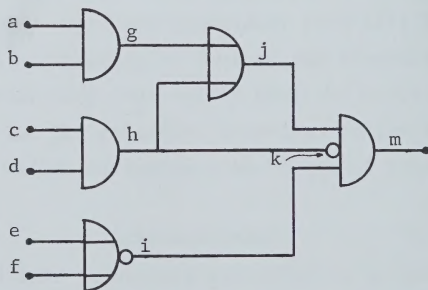


Fig. 5

		Lines												
Tests		a	b	c	d	e	f	g	h	i	j	k	m	
	1	1	1					1						
	2	0	1					0						
	3	1	0					0						
	4			1	1			1						
	5			0	1			0						
	6			1	0			0						
	7					0	0		1					
	8					0	1		0					
	9					1	0		0					
10								0	0		0			
11								0	1		1			
12								1	0		1			
13								1			0			
14								0			1			
15									1	1	1	1		
16									0	1	1	0		
17									1	0	1	0		
18									1	1	0	0		

Fig. 6

Tests	Lines a b c d e f g h i j k m											Comments
1	1	1	0	1	0	0	1	0	1	1	1	
2	0	1	1	0	0	0	0	0	1	0	1	
3	1	0	0	1	0	0	0	0	1	0	1	
4a	0	1	1	1	0	0	0	1	1	1	0	k value contradicts
4b	1	1	1	1	0	0	1	1	1	1	0	
5a	1	0	0	1	0	0	0	0	1	0	1	not valid since independent of h
5b	1	1	0	1	0	0	1	0	1	1	1	
6a	0	1	1	0	0	0	0	0	1	0	1	not valid since independent of h
6b	1	1	1	0	0	0	1	0	1	1	1	
7				0	0			1	1	1	1	covered by test 1
8	1	1	0	1	0	1	1	0	0	1	1	
9	1	1	1	0	1	0	1	0	0	1	1	
10							0	0	1	0	1	covered by test 2
11							0	1	1	1	1	undetectable fault
12							1	0	1	1	1	covered by test 6b
13							1	1	1	0	0	covered by test 4b
14							0	1	1	1	1	covered by test 1
15							1	1	1	1	1	covered by test 1
16							0	1	1	0	0	covered by test 8
17							1	0	1	0	0	covered by test 2
18							1	1	0	0	0	covered by test 4b

Fig. 7

ROBUST DETECTION PROCEDURES FOR SIGNALS OF KNOWN FORM IN NEARLY GAUSSIAN NOISE

R. D. MARTIN¹ and S. C. SCHWARTZ²
Princeton University, Princeton, New Jersey 08540

ABSTRACT

Procedures are given for the detection of signals of known form in nearly Gaussian noise. The tests allow for arbitrary small deviations in the statistics of the assumed model, while guaranteeing a specified level of performance. For the case of a constant signal in nearly Gaussian noise, the test procedure involves soft limiting of the observed data, and is seen to lie between the parametric (sample mean) test and the non-parametric sign test. This detector is called a limiter detector. Comparisons on the basis of false-alarm rate, power and A.R.E. demonstrate the advantages of the robust procedure.

For the case of time-varying signals, two detectors, which involve the use of a limiter, are considered. The first detector, the correlator-limiter, is robust in a min-max sense when the signal level is known. The second detector, the limiter-correlator, has the interesting property that the A.R.E. with respect to the correlator is the same as the A.R.E. of the limiter detector with respect to the sample mean.

INTRODUCTION

In recent years considerable work has been done in the area of non-parametric detection. An obvious shortcoming of the use of non-parametric tests in many situations is that the assumed family of possible underlying distributions is too broad. For example, if one suspects that the underlying distribution is nearly Normal in some sense, with zero median, then one would expect to pay a stiff penalty for assuming that the family consists of all zero median distribution functions. The penalty paid is to be expected from the rather severe data reduction involved in non-parametric procedures, and is reflected in the low A.R.E.'s (asymptotic relative efficiency) obtained when the distribution is actually Gaussian.

On the other hand, the parametric approach, with the usual assumption of Gaussian noise, also has obvious shortcomings. It is known that an optimum statistical procedure based on the Gaussian assumption may perform poorly for seemingly mild deviations from the Gaussian model [2].

It therefore seems desirable to establish tests which lie somewhere between parametric tests and non-parametric tests. Such tests have been described by statisticians as robust tests. Generally speaking, a robust test is one whose performance does not deteriorate too seriously for small deviations from a nominal model. One does not expect a robust test to perform as well as the optimal test based on the nominal model when the distribution fits the model exactly; however, it should perform

¹Research supported in part by National Science Foundation Grant NSF-GK-1439 and AFRADC under contract AF-F30602-68-C-0078.

²Research supported by National Science Foundation Grant NSF-GK-1532.

nearly as well. Also, the robust test should perform better than the parametric test for a large subset of the distributions which are near (according to some distance measure) to the nominal model. The test should generally perform better than its non-parametric counterpart.

In this paper we investigate the structure and performance of robust detectors for signals of known form in additive nearly Gaussian noise. By nearly Gaussian noise we mean that the distribution which describes the actual situation may be anywhere in a well-defined neighborhood of the assumed Gaussian distribution. In particular, we use the neighborhood defined by the mixture, or ϵ -contaminated, model. This model is given by:

$$F(x) = (1-\epsilon)N(\mu, \sigma) + \epsilon H(x), \quad 0 \leq \epsilon < 1 \quad (1)$$

where $N(\mu, \sigma)$ is the Normal c.d.f. with mean μ and standard deviation σ , and $H(x)$ is an arbitrary distribution, reflecting possible differences between the theoretical and actual situations.* This is one of the models of uncertainty used by Huber [1], who has provided the theoretical framework upon which the robust procedure is based.

Equation (1) defines a very broad family of distributions which are, in some sense, nearly Normal for small ϵ . Thus, if a detector is robust for the family defined by (1), we may feel confident in using the detector in situations for which it is known that the noise is nearly Normal but for which no detailed knowledge of the distribution is available.

In this paper the following problems will be considered:

$$x_i = \theta + n_i, \quad i=1, 2, \dots, N \quad (2)$$

where θ is a constant signal and the n_i are independent, identically distributed, nearly Normal noise samples.

$$x_i^j = \theta s_i + n_i^j, \quad i=1, 2, \dots, M; \quad j=1, 2, \dots, N \quad (3)$$

where the s_i are known and the n_i^j are the same as in (2). For $N > 1$ the problem is that of detecting a train of pulses, the $\{s_i\}$ set comprising the samples of an individual pulse. For these problems we consider detection when the signal amplitude is known ($H(\theta=0)$ against $K(\theta=\theta_0 > 0)$) and detection when the signal amplitude is unknown ($H(\theta=0)$ against $K(\theta > 0)$). For all of these problems the robust detector involves the use of a soft limiter.

CONSTANT SIGNAL IN NOISE

Constant Signal of Known Amplitude

We first consider the simple problem

$$H: x_i = n_i \quad K: x_i = \theta + n_i \quad i=1, 2, \dots, N$$

where $\theta > 0$ is known and where the n_i are independent, identically distributed, and $n_i \approx N(0, 1)$, i.e., the noise distribution is nominally the unit Normal distribution. The choice of $\sigma=1$ is only a matter of convenience. Huber's results [1] may be applied directly to this problem.

Huber's Results. With (χ, θ) a measurable space, P_0 and P_1 are assumed to be two distinct probability measures on it,

*Such a model, with $H(x) = N(\mu, \sigma\epsilon)$, was considered by Tukey in connection with the robust estimation of location as early as 1949 [2].

having densities p_0 and p_1 with respect to the measure $\mu = P_0 + P_1$. The assumed simple hypothesis P_0 and simple alternate P_1 are blown up into composite hypothesis by using the mixture model as in (1):

$$\theta_i = \{Q_i | Q_i = (1-\epsilon)P_i + \epsilon H_i, H_i \in \mathcal{H}\}, \quad i=0,1 \quad (4)$$

with $0 \leq \epsilon < 1$ and $\mathcal{H}$ representing the class of all probability measures on $(\chi, \mathcal{B})$. Let Q_i' represent an element of θ_i . Let $\underline{x} = (x_1, \dots, x_N)$ be a set of N independent observations from χ and let $\phi(\underline{x})$ represent a probability ratio test between some pair (Q_0', Q_1') . Also, let $R(Q_i', \phi)$ represent the risk associated with the pair (Q_i', ϕ) where the Q_i' in this pair is any element of θ_i , and $R(Q_i, \phi) = L_i E_{Q_i'}(\phi)$. Huber has solved the problem:

$$\min_{\phi} \max_{Q_1'} R(Q_1', \phi) \text{ subject to } \sup_{Q_0'} R(Q_0', \phi) \leq \alpha \quad (5)$$

We restate his solution (in slightly different form). Let Q_0 and Q_1 be given in terms of their densities by:

$$\begin{aligned} q_0(x) &= \begin{cases} (1-\epsilon)p_0(x) & p_1(x)/p_0(x) < c'' \\ \frac{1}{c'+1}(1-\epsilon)p_1(x) & p_1(x)/p_0(x) \geq c'' \end{cases} \\ q_1(x) &= \begin{cases} (1-\epsilon)p_1(x) & p_1(x)/p_0(x) > c' \\ c'(1-\epsilon)p_0(x) & p_1(x)/p_0(x) \leq c' \end{cases} \end{aligned} \quad (6)$$

The numbers c' and c'' have to be determined such that q_0 and q_1 are probability densities, i.e., c' and c'' must satisfy the equations:

$$\begin{aligned} P_1[p_1/p_0 > c'] + c'P_0[p_1/p_0 \leq c'] &= \frac{1}{1-\epsilon} \\ P_0[p_1/p_0 < c''] + \frac{1}{c'+1}P_1[p_1/p_0 \geq c''] &= \frac{1}{1-\epsilon} \end{aligned} \quad (7)$$

Equations (7) have unique solutions provided $0 \leq \epsilon < 1$. Furthermore, the relation $0 \leq c' < 1 < c'' \leq \infty$ is satisfied provided that ϵ is sufficiently small. Let ϕ be a probability ratio test between Q_0 and Q_1 . Then we have the

THEOREM (Huber):

The probability ratio test ϕ between q_1 and q_0 solves the problem (5) provided that ϵ is sufficiently small. The test ϕ is given by:

$$\phi(\underline{x}) = \begin{cases} 1, & T_n(\underline{x}) > C \\ a, & T_n(\underline{x}) = C \\ 0, & T_n(\underline{x}) < C \end{cases} \quad \text{where } T_n(\underline{x}) = \prod_{i=1}^N T(x_i) \text{ and}$$

$$T(x) = \begin{cases} c' & p_1/p_0 \leq c' \\ p_1/p_0 & c' < p_1/p_0 < c'' \\ c'' & p_1/p_0 \geq c'' \end{cases}$$

The test ϕ is then robust in a min-max sense, with minimum power $1-R(Q_1, \phi)$ and a maximum false alarm rate of $\alpha = R(Q_0, \phi)$. The pair (Q_0, Q_1) is said to be least favorable. The question as to whether or not the test has the desired robustness qualities mentioned in the introduction remains to be determined by actual

evaluation of the detector performance.

Some comments need to be made concerning Huber's model (4). Observe that the model has the characteristic that the measures under the alternate are not constrained to be translated versions of the hypothesis measures. This results in an overlap (i.e., at least one distribution will be contained in both classes) of θ_0 and θ_1 unless ϵ is sufficiently small. This is the reason for the requirement of small ϵ in the theorem. For our problems (2) and (3) this can be stated as follows: For fixed ϵ , θ_0 and θ_1 will overlap unless $\theta > \theta_{\min}(\epsilon)$. Consequently Huber's results do not hold in the small signal region.

Detector Structure. We now apply Huber's results to problem (2) when testing $H(x_i = n_i)$ against $K(x_i = \theta + n_i)$. The densities p_0 and p_1 , corresponding to H and K , respectively, are:

$$p_0(x) = K \exp(-x^2/2) \quad p_1(x) = K \exp[-(x-\theta)^2/2]$$

An individual term of the probability ratio is given by

$$p_1/p_0 = p_1(x)/p_0(x) = \exp(x\theta - (\theta^2)/2)$$

From (6), each term is to be limited at c' and c'' which is equivalent to limiting the logarithm of the individual terms at $\log c'$ and $\log c''$. With $\log c' = a'$ and $\log c'' = a''$ the test becomes:

$$\phi(\underline{x}) = \begin{cases} 1 & \gamma_N(\underline{x}) > C \\ a & \gamma_N(\underline{x}) = C \\ 0 & \gamma_N(\underline{x}) < C \end{cases} \quad \gamma_N(\underline{x}) = \sum_{i=1}^N \gamma(x_i)$$

$$\gamma(x) = \begin{cases} a' & x\theta - \frac{\theta^2}{2} \leq a' \\ x\theta - \frac{\theta^2}{2} & a' < x\theta - \frac{\theta^2}{2} < a'' \\ a'' & x\theta - \frac{\theta^2}{2} \geq a'' \end{cases}$$

The simplest detector structure is realized by passing the data samples, x_i , through a limiter, $\psi_{AL}^{AU}(\cdot)$, having values $AL = a'/\theta + \theta/2$ and $AU = a''/\theta + \theta/2$ and then summing, as shown in Fig. 1.

Limiter Values. The limiter values are obtained by solving equations (7) which may be written in terms of a' , a'' and the unit Normal c.d.f., $\Phi(\cdot)$, as:

$$1 - \Phi(a'/\theta - \theta/2) + e^{a'} \Phi(a'/\theta + \theta/2) = \frac{1}{1-\epsilon} \quad (8)$$

$$\Phi(a''/\theta + \theta/2) + e^{-a''} [1 - \Phi(a''/\theta - \theta/2)] = \frac{1}{1-\epsilon} \quad (9)$$

It is easily seen, noting the symmetry of the unit Normal c.d.f., that if a'' is a solution of (9) then $-a''$ is a solution of (8). Thus, the limiter of Fig. 1, with values $AL = a'/\theta + \theta/2$ and $AU = a''/\theta + \theta/2$, is a limiter whose center of symmetry is located at $\theta/2$, thereby limiting extreme values of both hypothesis and alternate distributions to the same extent.

Let $f_\theta(a'')$ denote the left hand side of (9) and let ϵ be fixed. Since $f_\theta(a'')$ is decreasing in a'' , a positive value of

a'' can be a solution of (9) only if $f_{\theta}(0) > 1/(1-\epsilon)$ is satisfied. This condition is satisfied only if $\theta > \theta_{\min}(\epsilon)$, reflecting the fact that the families ϕ_0 and ϕ_1 will overlap if $0 \leq \theta \leq \theta_{\min}(\epsilon)$. Some typical values of $\theta_{\min}(\epsilon)$ are: $\theta_{\min}(0.12) = .03$; $\theta_{\min}(0.055) = .15$; $\theta_{\min}(0.138) = .40$.

Consideration of the solution of $f_{\theta}(a'') = 1/(1-\epsilon)$ under various limiting conditions leads to the following conclusions. As $\theta \rightarrow \theta_{\min}$ for fixed ϵ (or $\epsilon \rightarrow \epsilon_{\max}$ for fixed θ), $a'' \rightarrow 0$ and the test becomes the sign test. As $\epsilon \rightarrow 0$ for fixed θ , $a'' \rightarrow \infty$ and the test is the sample mean test, as expected. As $\theta \rightarrow \infty$, a'' is the solution of $\exp(-a'') = \epsilon/(1-\epsilon)$ and the limiter values, AL and AU, approach $\theta/2$. In this case the test does not become the sample mean test, as might be hoped for. This simply reflects the fact that the problem is not parametric when $\epsilon > 0$, no matter how large θ .

Constant Signal of Unknown Amplitude

The choice of limiter values in the previous section depends explicitly upon the actual value of θ , where θ was assumed to be known. However, we are frequently interested in detecting a constant signal of unknown amplitude and we need to determine appropriate limiter values for this case.

Let ϕ_{θ_d} denote the test based on a particular $\theta_d > \theta_{\min}(\epsilon)$. For this test we shall find a sharpest lower bound on the power, which we denote by $B_{\theta_d}(\theta)$, for all $\theta > 0$.

The test $\phi_{\theta_d}(\underline{x})$ is based on

$$\gamma_{N_{\theta_d}}(\underline{x}) = \sum_{i=1}^N \gamma_{\theta_d}(x_i), \text{ where } \gamma_{\theta_d}(x) = \log[q_{\theta_d}(x)/q_0(x)]$$

Let x_0 be defined by $\gamma_{\theta_d}(x_0) = a'$, and let $Q_{\theta} = (1-\epsilon)P_{\theta} + \epsilon\tilde{H}$ where $\tilde{H}$ is a probability measure which places all of its mass at x_0 , $Q_{\theta} = (1-\epsilon)P_{\theta} + \epsilon\tilde{H}$, $\tilde{H} \in \mathcal{H}$, and $\theta \in \mathcal{R}^1$. Then we have the

LEMMA:

$$B_{\theta_d}(\theta) = E_{Q_{\theta}}(\phi_{\theta_d}) \leq E_{Q_{\theta}}(\phi_{\theta_d}) \quad (10)$$

Proof:

See [3].

For our problem of constant signal in nearly Gaussian noise, the density corresponding to Q_{θ} is $q_{\theta}(x) = (1-\epsilon)\phi_{\theta}(x) + \epsilon\delta(x - AL_{\theta_d})$, where $\phi_{\theta}(x)$ is the Normal density function with mean θ and unity variance and $\delta(\cdot)$ is the unit impulse, i.e., all of the ϵ -component is located at the lower limiter value.

It is important to note that the test ϕ_{θ_d} is not necessarily a probability ratio test for any pair $(Q'_{\theta}, Q''_{\theta})$ when $\theta \neq \theta_d$. For this reason ϕ_{θ_d} may result in a better lower bound on the power for a given θ than the lower bound obtained by designing the test for that θ .

The question now arises as to what value of θ_d one should actually use. What θ_d gives the best overall performance in some sense? It appears that there is no analytical solution to this question, i.e., there exists no nice ordering relation for the tests based on various θ_d . Therefore, the selection of θ_d , which yields the limiter values, must be based on numerical evaluation of detector performance for various θ_d .

Power and False Alarm. The test statistic $\gamma_{N_{\theta_d}}(\underline{x}) = \sum_{i=1}^N \gamma_{\theta_d}(x_i)$,

with each x_i distributed according to Q_0 , generally has a complicated non-Normal distribution for finite N . However, the statistic is asymptotically Normal and for moderately light limiting and moderate signal levels the test statistic will be essentially Normal for reasonably small sample sizes. Hence, we make calculations for moderately large N , assuming that $y_{N\theta_d}$ is Normal. We present the following specific results. With $\epsilon=.01$, $\alpha=.005$ and $N=200$ Fig. 4 gives: $B_{\theta_d}(\theta)$ for various θ_d ; the power of ϕ_{θ_d} when the distribution is actually Gaussian; the sample mean power and false alarm for the Gaussian and for the two ϵ -contaminated densities,

$$q_{\theta}(x) = \frac{(1-\epsilon)}{\sqrt{2\pi}} e^{-\frac{(x-\theta)^2}{2}} + \frac{\epsilon}{\sqrt{2\pi} \sigma_{\epsilon}} e^{-\frac{(x-\theta)^2}{2\sigma_{\epsilon}^2}}, \quad \sigma_{\epsilon}=10, 30.$$

The design value $\theta_d=.06$ yields a nearly best minimum power curve. The corresponding limiter values are $AL=-.56, AU=+.62$. It is to be noted that although the $\theta_d=.06$ curve is not uniformly best, it falls only slightly below the $\theta_d=.03$ curve in the small signal region. These curves reflect the intuitive feeling that moderate limiting will result in a best robust test since harder limiting results in too much data reduction and lighter limiting results in too much sensitivity to tail behavior of the distribution. Actually, the $\theta_d=.10$ curve results in nearly as good a performance; the corresponding limiter values are $AL=-.82$ and $AU=+.92$. This suggests that, in general, the minimum power will be about the same for $1.2 \leq AU-AL \leq 1.8$.

Note that the behavior of the sample mean detector for the ϵ -contaminated models is very poor in terms of false alarm rate, e.g., for $\sigma_{\epsilon}=10$ the false alarm has increased by a factor of 7, whereas the robust detector has $\alpha \leq .005$. This extremely good protection against false alarm variability is obtained at a moderate cost in power.

A.R.E. We now consider the performance of the detector of Fig. 1, which we refer to as a limiter, in terms of its A.R.E., $e_{l,m}$, with respect to the sample mean, and shall make some specific comparisons with the A.R.E., $e_{s,m}$, of the sign detector with respect to the sample mean.

By restricting attention to alternate densities which are translated versions of hypothesis densities, and imposing some mild regularity conditions, it may be shown that [3]:

$$e_{l,m} = \left[\int_{AL}^{AU} q_0(x) dx \right]^2 (\sigma_n^2) / (\sigma_l^2) \quad \text{where } \sigma_l^2 = \text{var}\{l_{AL}^{AU}(x) | q_0(x)\}. \quad (11)$$

For symmetrical distributions having a density function which is continuous at the origin, the A.R.E. of the sign detector with respect to the sample mean is given by [4]:

$$e_{s,m} = 4 \sigma_n^2 q_0^2(0)$$

$e_{l,m}$ is evaluated for a limiter with values of $AL=-.56$ and $AU=+.62$ (corresponding to $\epsilon=.01$ and $\theta_d=.06$ in the previous section). The model of nearly Normal noise for which $e_{l,m}$ and $e_{s,m}$ are evaluated is:

$$q_0(x) = \frac{(1-\epsilon)}{\sqrt{2\pi}} e^{-x^2/2} + \frac{\epsilon}{\sqrt{2\pi} \sigma_{\epsilon}} e^{-(x^2)/(2\sigma_{\epsilon}^2)}, \quad \epsilon=.01, 1 \leq \sigma_{\epsilon} \leq 10$$

The results, shown in Fig. 5, clearly indicate the robustness of the limiter detector.

SIGNAL OF KNOWN FORM

In this section the problem (3) of detecting a signal of known form in additive nearly Gaussian noise is discussed.

Known Signal Amplitude

When the signal amplitude is known we have the problem

$$H: x_i^j = n_i^j \quad K: x_i^j = \theta s_i + n_i^j \quad i=1,2,\dots,M; j=1,2,\dots,N$$

where $\theta > 0$ is known. Huber's work applies directly only to the case of constant signal in noise. However, with some rather straightforward modifications, his work may be extended to cover this problem. One of the possible forms of the resulting detector is based on [3]:

$$\sum_{i=1}^M \sum_{j=1}^N \frac{AU_i}{AL_i} (s_i x_i^j) \geq C; \quad AL_i = \frac{a_i'}{\theta} + \frac{\theta s_i^2}{2}, \quad AU_i = \frac{a_i''}{\theta} + \frac{\theta s_i^2}{2} \quad \text{where}$$

$a_i' = -a_i''$ and a_i'' is the solution of

$$e \left(\frac{a_i''}{\theta |s_i|} + \frac{\theta |s_i|}{2} \right) + e^{-a_i''} \left[1 - e \left(\frac{a_i''}{\theta |s_i|} - \frac{\theta |s_i|}{2} \right) \right] = \frac{1}{1-\epsilon} \quad (12)$$

This detector, which will be referred to as the correlator-limiter, is shown in Fig. 2.

Thus, the robust detector performs a sample-wise correlation followed by limiting, rather than limiting followed by correlation. Since the variable component of each term, $x_i^j s_i$, of the test statistic is $n_i^j s_i$ and since the effect of the limiter is to prevent unexpected extreme values of n_i^j from upsetting the statistic and ensuing decision, one expects, intuitively, that the correlator-limiter may be a robust detector.

Unknown Signal Amplitude

The evaluation of the detector in terms of power and false alarm requires calculations based on the choice of a specific signal set, $\{s_i\}_{i=1}^M$. Such an evaluation would be required in order to determine a good design value for θ_d . We forego such calculations and note that the results for constant signal of unknown amplitude suggest that the detector of Fig. 2, with $AL_i = AL = -.8$ and $AU_i = AU = +.9$, is a good robust detector for $\epsilon = .01$.

Detector Performance - A.R.E.

Using the same assumptions that were used in deriving the expression (11) for $e_{l,m}$, it may be shown that $e_{cl,c}$, the A.R.F. of the correlator-limiter with respect to the conventional correlator [5] is given by [3]:

$$e_{cl,c} = \frac{\left[\sum_{i=1}^M s_i^2 \int_{AL/s_i}^{AU/s_i} q_0(x) dx \right]^2}{\sum_{i=1}^M \frac{AU}{AL} \text{var}\{t(s_i x) | q_0(x)\}} \cdot \frac{\sigma_n^2}{\sum_{i=1}^M s_i^2} \quad (13)$$

Once a signal set has been specified, $e_{cl,c}$ may be evaluated for various densities, $q_0(x)$.

We now consider the closely related alternate detector structure of Fig. 3, which has the same intuitively appealing feature as does the correlator-limiter, namely, extreme values of x_i^j arising from the noise component are suppressed. We call this detector the limiter-correlator. It may be shown that the A.R.E., $e_{LC,C'}$, of the limiter-correlator with respect to the correlator is [3]

$$e_{LC,C} = \left[\int_{AL}^{AU} q_0(x) dx \right]^2 \frac{\sigma_n^2}{\sigma_s^2} \quad (14)$$

This is the same expression as that for $e_{LC,M}$, the A.R.E. of the limiter with respect to the sample mean (c.f., (11)). Thus, the limiter-correlator (for a signal set $\{s_i\}_{i=1}^M$) has the same performance, and hence the same degree of robustness, as the limiter (for a constant signal) in terms of A.R.E.

COMMENTS

The correlator-limiter detector is the min-max solution analogous to the limiter detector. It is conjectured that the correlator-limiter detector has robustness features on a par with those of the limiter detector. The closely related (structurally) limiter-correlator is as robust as the limiter detector in terms of A.R.E. for at least two non-Normal models [3]. An open question is that of determining which of these two detectors is most robust in various situations, i.e., is one form better than the other in the small signal region?

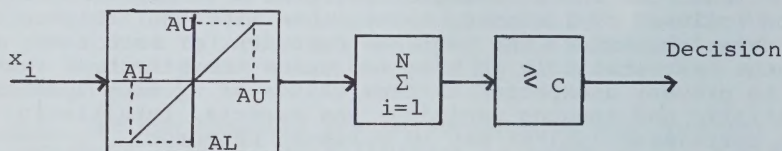


Fig. 1

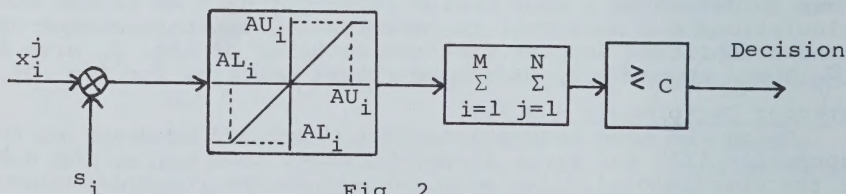


Fig. 2

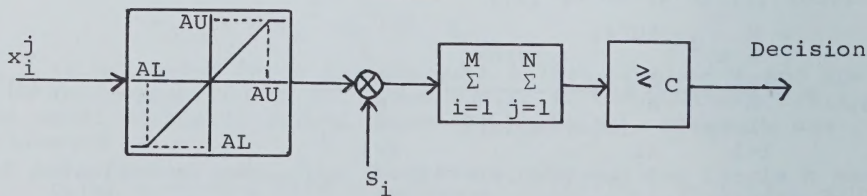
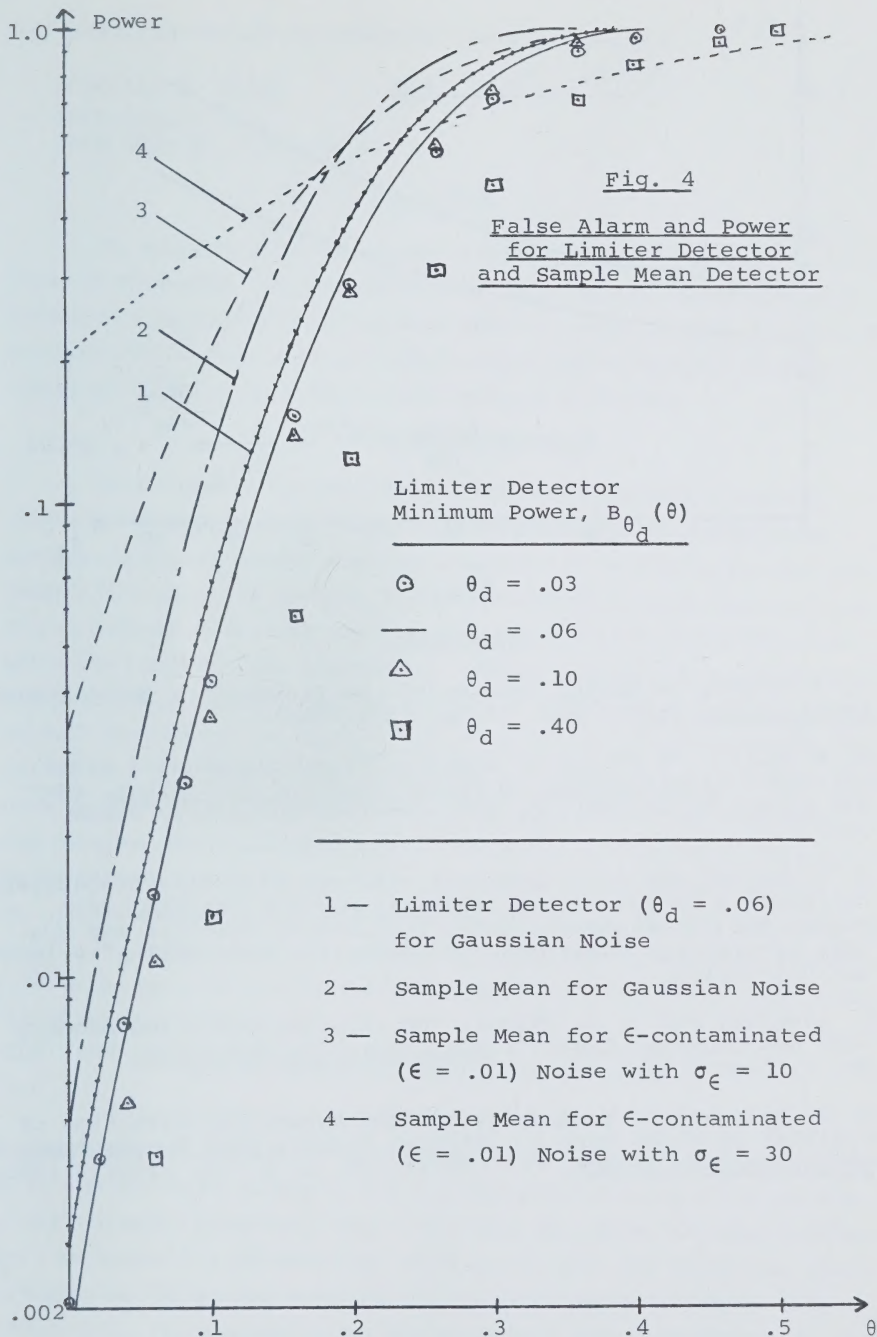


Fig. 3



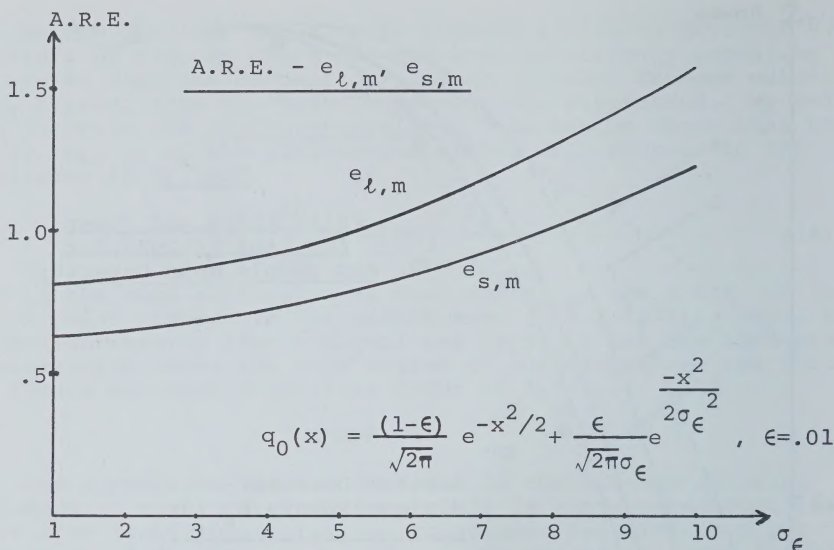


Fig. 5

REFERENCES

1. P. J. Huber, "A Robust Version of the Probability Ratio Test", Annals Math. Stat., Vol. 36, p. 1753, 1965.
2. J. W. Tukey, "A Survey of Sampling From Contaminated Distributions", Contributions to Probability and Statistics, (Harold Hotelling Volume), p. 448, Stanford University Press, 1960.
3. R. D. Martin and S. C. Schwartz, "Robust Detection Procedures for Signals of Known Form in Nearly Gaussian Noise", Tech. Report No. 25, September 1968, Information Sciences and Systems Laboratory, Department of Electrical Engineering, Princeton University.
4. M. Kanefsky and J. B. Thomas, "On Polarity Detection Schemes with Non-Gaussian Inputs", Jour. Franklin Institute, Vol. 280, No. 2, August 1965.
5. R. F. Daly and C. K. Rushforth, "Non-Parametric Detection of a Signal of Known Form in Additive Noise", IEEE Transactions on Information Theory, Vol. IT-11, No. 1, p. 20, January 1965.

AN ANALYSIS OF AN AUTOMATIC THRESHOLD UNIT

THOMAS E. EGER

HRB-Singer, Inc.

State College, Pennsylvania

ABSTRACT

An analysis of an automatic threshold device of the feed forward class is presented. Of primary interest is the minimization of the threshold time uncertainty resulting from additive noise. A generalized feed forward unit is characterized mathematically and a specific model is analyzed. Finally, an experimental model is described.

I. INTRODUCTION

In this paper, we are concerned with an automatic threshold device of the feed forward class. The feed forward concept involves system parameter control which is a function of current rather than past input information. In general, an input waveform to a feed forward system is delayed or in some other manner stored while certain decisions are made relative to its constitution. System parameters are adjusted based on these decisions and the delayed or stored waveform is passed through the system. Application of this concept to an automatic threshold unit results in a device which thresholds at a prescribed percentage of the input peak pulse amplitude for each pulse regardless of its amplitude. In this manner, the undesirable shift of threshold time caused by input amplitude changes (normally encountered with fixed thresholders) is eliminated. An automatically variable threshold unit must establish the peak pulse amplitude, reliably set a threshold level using this information, and threshold on the input pulse when the input signal amplitude exceeds the established fraction of the peak.

II. MODEL DEVELOPMENT AND ANALYSIS

An example of an automatic threshold unit is shown in Figure 1. An input pulse of amplitude E_p is applied to the input of a delay line and to a peak amplitude estimator whose output level is equal to the negative of the last pulse amplitude, $-E_p$. The delay line stores the input until the peak estimator has measured the pulse amplitude. The delay line output is multiplied by two and summed with the peak estimator output. For a constant E_p , the summer output is a pulse train with base line at $-E_p$ and 50% level at 0 volt. Thus, a zero crossing detector is used to select the half amplitude point.

Both the estimation of the peak amplitude and the thresholding process are perturbed by noise. It is impossible to know exactly the pulse amplitude if noise is added to the pulse. Similarly, the threshold time is a random variable if noise is added to the leading edge of the input pulse.

The effect of noise added to the leading edge of a pulse and fed to a fixed thresholder is given by,

$$\sigma_t = \frac{\sigma_N}{M} \quad (1)$$

where σ_t is the standard deviation of the random variable describing the time jitter, σ_N is the standard deviation of the noise, and M is the slope of the pulse in the neighborhood of the threshold level. Now if the threshold level is not constant, as is the case when the peak amplitude is estimated to set the threshold level, the noise enters the picture a second time. If the peak value estimate of the pulse amplitude has a standard deviation σ_p then

$$\sigma_t = \sqrt{\left(\frac{\sigma_N}{M}\right)^2 + \left(\frac{\sigma_p}{2M}\right)^2} \quad (2)$$

The factor of 2 enters in since the threshold level is half the peak amplitude. The first term under the radical results from the noise added to the leading edge of the input pulse, the second from the effect of the noise in the peak estimation process. It is convenient to introduce an improvement factor β which relates the standard deviation of the noise at the output of the peak estimator to that at the input,

$$\beta = \left(\frac{\sigma_p}{\sigma_N}\right)^2 \quad (3)$$

The rms time error then becomes

$$\sigma_t = \frac{\sigma_N}{M} \sqrt{1 + \beta/4} \quad (4)$$

One of the major design criterion for an automatic threshold device must be the minimization of σ_t . Excluding any averaging which might be done by making use of multiple threshold units, the most fruitful area of investigation is that concerned with the peak estimation process. The method chosen for the peak estimator is analog integration. That is, the peak amplitude is specified as the time integral of the pulse waveform.

The performance of the integrating peak estimator is developed by obtaining an expression for the rms uncertainty of the integrator output. Essentially, the problem reduces to that of estimating the mean

value of a random process by integrating over an interval T_I . In the present case, the mean value is the pulse amplitude. Lee⁽²⁾ derives the variance of the output of such an integrator as,

$$\sigma_p^2 = \int_{-T_I}^{T_I} \frac{T_I - |\tau|}{T_I^2} (R(\tau) - m^2) d\tau \quad (5)$$

where $R(\tau)$ is the autocorrelation function of the noise input and m is its mean. The mean m , is zero if the noise added to the pulse is assumed to have no dc component. Then, since $R(\tau)$ is an even function of τ , σ_p^2 reduces to

$$\sigma_p^2 = 2 \int_0^{T_I} \frac{T_I - \tau}{T_I^2} R(\tau) d\tau \quad (6)$$

The autocorrelation function may be calculated as the Fourier transform of the power density spectrum of the noise $S(\omega)$. Once $R(\tau)$ is available, σ_p^2 may be calculated. σ_p^2 represents the noise power out of the integrator and it will be recalled that $R(0)$ is the noise power into the integrator. Thus, the ratio of σ_p^2 to $R(0)$ represents the improvement (reduction) in noise power at the integrator output relative to the noise power at the input. This improvement factor has been denoted as β .

Two cases are considered for the integrating peak estimator. The first case represents the input noise and signal as having been band limited by a Gaussian-shaped filter. The second case assumes a rectangular filter shape restricts the noise and signal spectra. These cases yield different autocorrelation functions and different values of waveform slope. A third case is included for the sake of comparison; this involves replacing the integrating peak estimator with a simple RC filter.

When the input noise is shaped by a filter with Gaussian transfer characteristic,

$$H(j\omega) = \text{EXP}(-\omega^2/2\omega_0^2) \quad (7)$$

the resulting autocorrelation function is,

$$R_G(\tau) = \frac{N \omega_0}{2\sqrt{\pi}} \text{EXP}(-(\tau\omega_0/2)^2) \quad (8)$$

N is the noise power density of the input noise and ω_0 is a measure of the filter bandwidth. β may then be calculated as,

$$\beta_G = \frac{2\sqrt{2}}{\omega_o T_I} \left\{ \sqrt{2\pi} \left[\Phi\left(\frac{\omega_o T_I}{2}\right) - \frac{1}{2} \right] + \frac{2}{\omega_o T_I} \left[\text{EXP}\left(-\left(\frac{\omega_o T_I}{2}\right)^2\right) - 1 \right] \right\} \quad (9)$$

where

$$\Phi(X) = \frac{1}{2} \int_{-\infty}^X \text{EXP}(-y^2/2) dy \quad (10)$$

A plot of β as a function of $\mu = \omega_o T_I$ is given in Figure 2. When the slope of a pulse passed through this Gaussian shaped filter is evaluated at the fifty percent amplitude point, the following results,

$$\sigma_t = \frac{1}{\omega_o(S/N)} \sqrt{1 + \beta_{G/4}} \quad (11)$$

where (S/N) is the peak pulse amplitude to rms noise voltage ratio

The second case studied assumes a noise spectrum and a pulse spectrum band limited by a rectangular shaped filter. Thus for

$$H(j\omega) = 1 \text{ for } -\omega_c \leq \omega \leq \omega_c \quad (12)$$

$$= 0 \text{ elsewhere}$$

$$\text{then } R_R(\tau) = \frac{\omega_c N_o}{\pi} \left(\frac{\text{Sin}(\omega_c \tau)}{\omega_c \tau} \right) \quad (13)$$

$$\text{and } \beta_R = \frac{2}{\omega_c T_I} \text{Si}(\omega_c T_I) + \frac{2}{(\omega_c T_I)^2} (\text{Cos}(\omega_c T_I) - 1) \quad (14)$$

$$\text{where } \text{Si}(X) = \int_0^X \frac{\sin y}{y} dy \quad (15)$$

The functional relationship of β on $\mu = \omega_c T_I$ is presented in Figure 2. When the slope at the fifty percent amplitude level is taken into account, the value of σ_t is

$$\sigma_t = \frac{\pi}{\omega_c(S/N)} \sqrt{1 + \beta_{R/4}} \quad (16)$$

and again (S/N) is the peak signal to rms noise ratio.

Figure 3 compares the rms threshold jitter for these two cases. The integration time for this example is chosen as 100 μ s while $\omega_o = 5 \times 10^4$ and $\omega_c = 4.44 \times 10^4$. The different bandwidth is used so that equal noise power results in both cases.

The final case examined compares the ideal integrator with a simple RC filter. The value of β is defined as the ratio of the output noise power to the input noise power. The filter shape is,

$$H(j\omega) = \frac{1}{1 + j\omega RC} \quad (17)$$

Then,

$$\beta_F = \frac{P_o}{P_I} = \left(\frac{1}{T_I \omega_c} \right) \tan^{-1} \left(\frac{T_I \omega_c}{5} \right) \quad (18)$$

Where the input noise is assumed to have a rectangular shaped spectrum with width ω_c . In this computation the integration time is chosen as $5RC$. β_F is plotted in Figure 2 as a function of $\mu = \omega_c T_I$.

III. EXPERIMENTAL WORK

An experimental model of the automatic threshold unit is shown in Figure 4. The input signal is processed by logic which selects the center section of the pulse. The signal is simultaneously applied to the peak estimator and to a delay line.

The peak estimator is updated with each input pulse. The previous pulse amplitude is stored on capacitor C and the difference between this stored value and a new input value is integrated. Thus, when the logic instructs the peak estimator to operate, analog gate A is closed while analog gate B is opened.

The integrator output is negative of the input pulse amplitude ($-E_p$). It is summed with the delayed pulse which had been amplified by a factor of two. The integrator output represents the base line upon which the delayed pulse train rides. This base line always has the value $-E_p$ and the delayed pulse always has total height $2E_p$ so the 50% amplitude point is always at zero. Thus, a zero crossing detector is used as the threshold output.

IV. SUMMARY AND CONCLUSIONS

This paper has outlined the development of an automatically variable threshold unit. A mathematical formulation of the system was presented in a generalized manner so as to be of value for any threshold unit employing some peak estimating device in conjunction with a trigger circuit. The role of the low pass filter which shapes the signal and the noise spectrum has been examined and the threshold jitter advantage of the Gaussian shaped filter relative to the rectangular shaped filter was indicated. The relative values of β for the peak estimating techniques were presented and the total impact of these techniques in terms of σ_t was illustrated. With the analytical results which have been obtained and which have been verified at least partially by experimentation, it would appear that the limitation of practically obtainable accuracy in terms of

peak estimation process have been obtained. Finally, from the formulation presented, the relationship of bandwidth, integration time (ideal or RC), filter frequency shape, and signal to noise ratio to σ_t are readily available.

REFERENCES

1. M. Schwartz, Information, Transmission, Modulation and Noise, McGraw-Hill.
2. Y. W. Lee, Statistical Theory of Communications, Wiley.
3. W. Davenport and W. Root, Random Signals and Noise, McGraw-Hill.
4. T. Eger, Analysis of An Adaptive Threshold Unit, Interim Report 913.30, HRB-Singer, Inc.

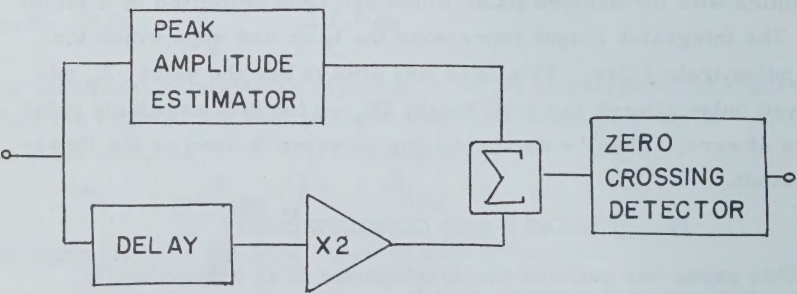


FIGURE 1 TYPICAL AUTOMATIC THRESHOLD

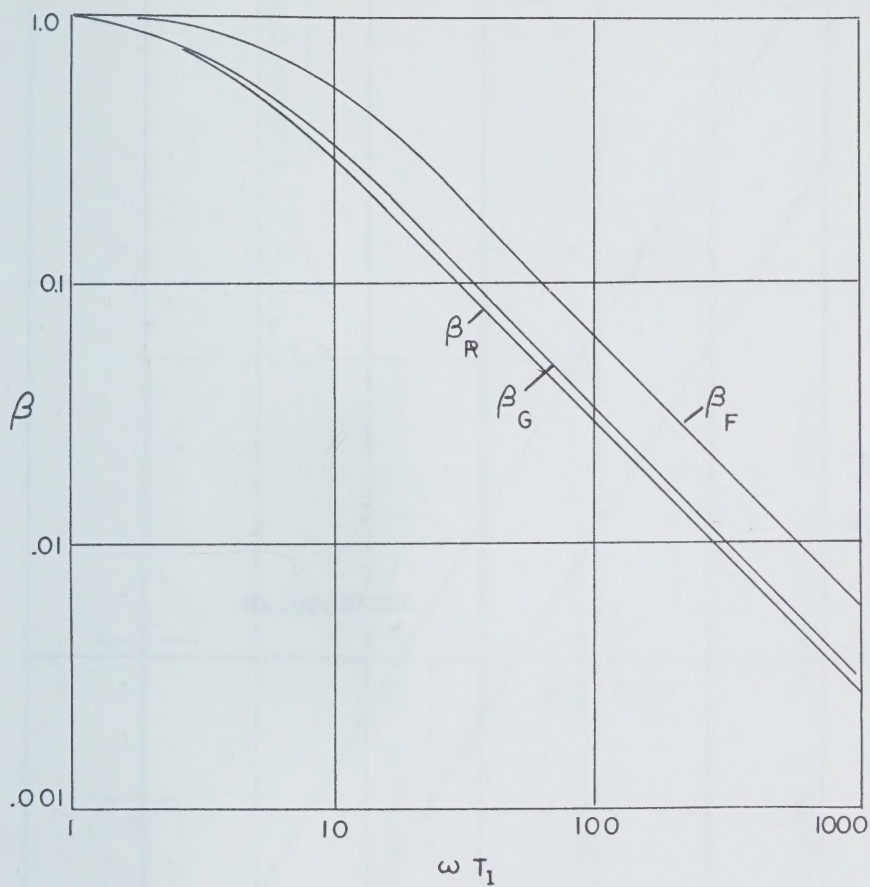


FIGURE 2 β vs. μ

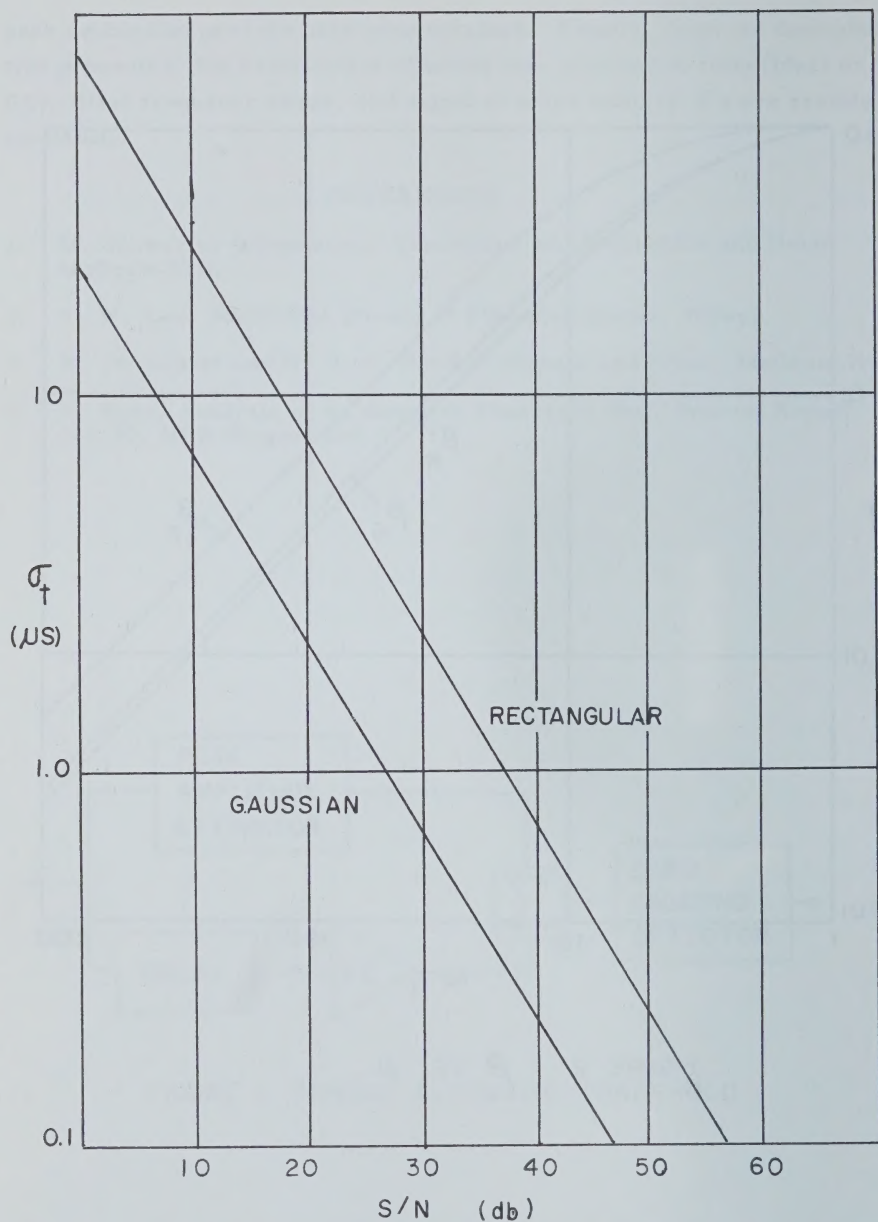


FIGURE 3 σ_t VS. S/N

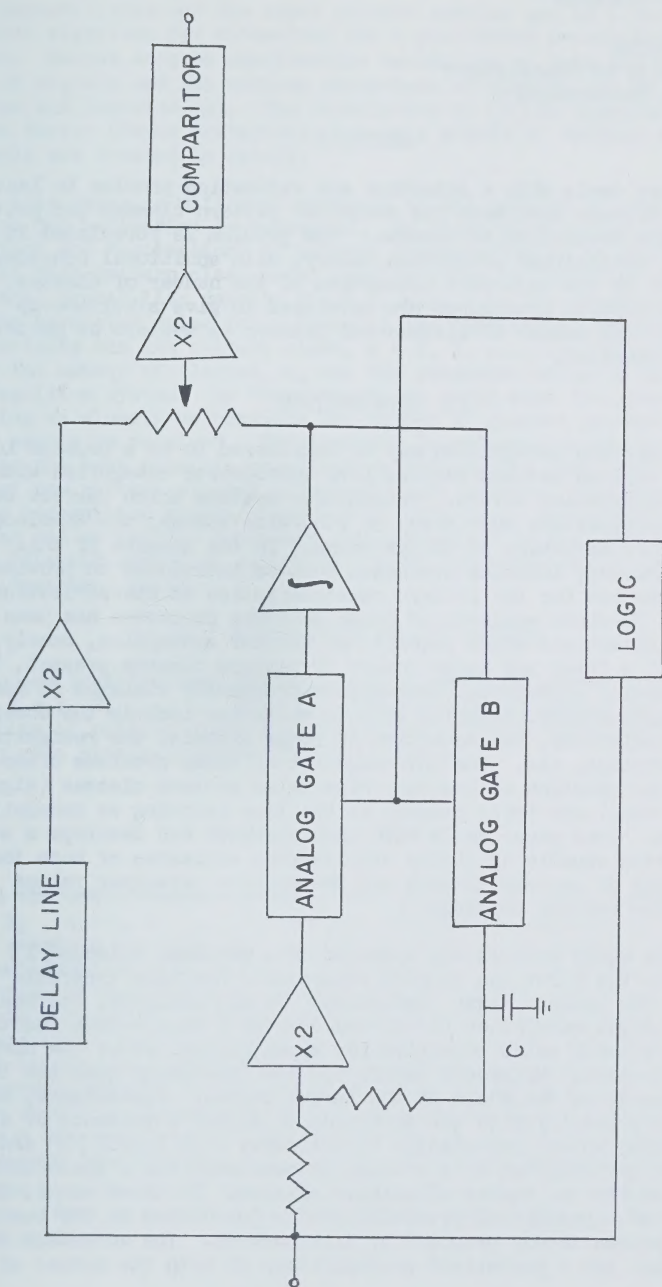


FIGURE 4 EXPERIMENTAL MODEL

SIMULTANEOUS DETECTION AND ESTIMATION OF PATTERN CHARACTERISTICS IN SELF-LEARNING SYSTEMS*

Y. T. CHIEN
University of Connecticut
Storrs, Connecticut

ABSTRACT

This paper deals with a detection and estimation problem in learning recognition systems when both the number of pattern classes and pattern parameters are assumed to be unknown. The problem is formulated in terms of the usual statistical estimation theory, with additional consideration of losses due to the incorrect assumption of the number of classes. Recursive estimation procedures are developed to give simultaneous estimates for the number of classes and pattern parameters as pattern samples are acquired.

I. INTRODUCTION

Optimal pattern recognition may be considered to be a problem in classifying various pattern samples into appropriate categories with minimum classification errors. Recognition systems which perform such optimal classifications must have, or otherwise assume, the knowledge about the characteristics of the patterns. In the absence of this knowledge, however, learning processes must be introduced to provide adequate estimates for the pattern characteristics as classification progresses. Previous analysis of these learning processes has been confined to situations which require one further assumption, namely, the assumption of a fixed and known number of pattern classes present. This further essential assumption, however, is frequently violated in many practical applications. Examples of this violation include the observation of nuclear radiations, the detection of radar signals, the recognition of human handwritings, etc. The investigation of these problems often results in the question of how many objects or pattern classes (signal, characters, etc.) are truly present at the time learning or recognition is performed. This paper deals with this question and develops a self-learning system capable of giving simultaneous estimates of both the unknown number of pattern classes and the unknown parameter values describing the various patterns.

Among the early workers who attacked this problem, Nilsson [1] apparently is the first one to have attempted a decision theoretic formulation for optimum radar resolution. In his analysis, the design of optimum target resolution is carried out for a square-loss function, modified to provide extra penalties for wrong guesses about the number of targets present. Nilsson's result has the limitation that the loss function depends on the shape of the target signal. Furthermore, his method deals primarily with the detection of signal's presence or absence. Several Russian works, represented by Bolohakov [6], Latysh [5] and others, have investigated this problem by obtaining the a posteriori probabilities for the number of pattern classes. In these works, the computation of a posteriori probabilities is formulated on the basis of the random points theory proposed by Stratonovich. The advantage of this method is that the a posteriori probabilities of both the number of

* This work was supported in part by a grant from the University of Connecticut Research Foundation.

pattern classes and the pattern parameters may be calculated from the a priori probabilities and the input pattern samples and as a result, a systematic algorithm for estimating the a posteriori probabilities is obtained. Recent work on simultaneous estimation of both the unknown number of signals and the unknown parameters of signals can be found in Middleton and Esposito [4]. The formulation in [4] is based on statistical decision theory (Bayes criterion), but only models of testing simple hypothesis are treated in detail.

In this paper, a different approach to the problem of simultaneous detection and estimation is presented. We first distinguish between "detection" and "estimation" of pattern characteristics in the following way. Let each pattern class under consideration be described by a conditional probability density $P(x/\theta_k)$, when x is the pattern sample and θ_k is the parameter value (e.g., mean of the conditional density) characterizing the k th pattern class, $k = 1, 2, \dots, m$. It is assumed that neither the number of classes, m , nor the parameter value θ_k is known to the recognition system. By "detection", we shall mean the process of announcing as closely as possible the number of classes present each time a pattern sample is received. "Estimation" is referred to the process of producing the best possible quantitative information about the parameter values of the pattern classes announced to be present. The purpose of the learning recognition system discussed in this paper is to perform the detection and estimation processes simultaneously so that the true state of affairs (classification of patterns) can be recognized in an optimal fashion described below.

II. FORMULATION OF THE PROBLEM

Suppose that for each pattern class the probability of its occurrence in P_k , $k = 1, 2, \dots, m$. Then it is reasonable to assume that the entire information of pattern samples is contained in the probability density function of a mixture distribution $P(x)$, where

$$P(x) = \sum_{k=1}^m P_k P(x/\theta_k) \quad (1)$$

Let X be the set of pattern samples consisting of a number of disjoint subsets X_k , $k = 1, 2, \dots, m$. Pattern samples are denoted by x . Each pattern sample x is considered a realization from the mixture $P(x)$. The class membership of x is assumed to be unknown. For each received sample x , two types of decision must be made automatically. The first decision is to make a guess on how many pattern classes are truly present (detection); the second is to decide for each x , in a best possible way, which pattern class it belongs to (estimation and classification). Let the loss due to the incorrect decision of the first type be denoted by $F(x, m, \hat{m})$. This is the loss when $\hat{m}$ is the guessed number of pattern classes and there are actually m classes present. Let the losses due to the incorrect decision of the second type be $G_k(x, \theta_1, \dots, \theta_m)$ when x is from the k th class. The total loss resulting from these two decisions is the sum of the two loss functions. From these two loss functions we define the corresponding expected risks:

$$R_1 = \int_X \left\{ \int_m F(x, m, \hat{m}) P(m/x) dm \right\} P(x) dx \quad (2)$$

R_1 is the expected risk due to the incorrect detection of the number of pattern classes. $P(m/x)$ is the a posteriori probability that m pattern classes are present given the learning pattern sample x . Similarly,

$$R_2 = \sum_{k=1}^{\hat{m}} \left\{ P_k \int_{X_k} G_k(x, \theta_1, \dots, \theta_{\hat{m}}) P(x/\theta_k) dx \right\} \quad (3)$$

Or, using equation (1) when the regions X_k are disjoint, we have

$$R_2 = \sum_{k=1}^{\hat{m}} \left\{ \int_{X_k} G_k(x, \theta_1, \dots, \theta_{\hat{m}}) P(x) dx \right\} \quad (4)$$

R_2 is the expected risk due to the misclassification of pattern samples x .

Now, introducing the characteristic function as in [3],

$$I_k(x, \theta_1, \dots, \theta_{\hat{m}}) = \begin{cases} 1 & \text{if } x \in X_k \\ 0 & \text{if } x \notin X_k \end{cases} \quad (5)$$

we can write the total risk in the following way

$$\begin{aligned} R &= R(\theta_1, \dots, \theta_{\hat{m}}, \hat{m}) \\ &= \int_X \left\{ \int_m F(x, m, \hat{m}) P(m/x) dm \right\} P(x) dx \\ &\quad + \sum_{k=1}^{\hat{m}} \int_X I_k(x, \theta_1, \dots, \theta_{\hat{m}}) G_k(x, \theta_1, \dots, \theta_{\hat{m}}) P(x) dx \end{aligned} \quad (6)$$

Or, briefly

$$\begin{aligned} R &= R(\theta_1, \dots, \theta_{\hat{m}}, \hat{m}) \\ &= E_x \left\{ \int_m F(x, m, \hat{m}) P(m/x) dm \right. \\ &\quad \left. + \sum_{k=1}^{\hat{m}} I_k(x, \theta_1, \dots, \theta_{\hat{m}}) G_k(x, \theta_1, \dots, \theta_{\hat{m}}) \right\} \end{aligned} \quad (7)$$

A desirable criterion to use in making the two decisions would be to require that R be minimized by simultaneous choice of an optimal value for the number classes present, $\hat{m}$, and a set of optimally-estimated parameter values for θ_k , $\hat{\theta}_k$, $k = 1, 2, \dots, \hat{m}$. Two observations with respect to this criterion should be made:

- (1) The problem stated here is one of learning the pattern characteristics (number of classes, pattern parameters) without the exact knowledge of the form of conditional probability distributions for the patterns. The pattern samples received are unclassified and hence no training sequence is assumed. A recognition system of this kind is referred as a self-learning recognition system, a term coined by Tsypkin [7].
- (2) The purpose of optimally estimating the pattern parameter θ_k is to produce optimal decision boundaries in such a manner that pattern samples may be classified with minimum error. In other words, we must find the regions X_k in X without knowing the class membership of x 's as they are acquired. It should be noticed that if the risk function depends explicitly only on the parameters θ_k , we can find the regions

X_k using the characteristic function I_k after finding the optimal estimates for θ_k .

III. SOLUTION OF THE PROBLEM

The minimization of R is based on the process of selecting simultaneously the optimal estimates for θ_k , $\hat{\theta}_k$, and the optimal number for m , $\hat{m}$, as each pattern sample is received. We assume that the pattern parameters θ_k , $k = 1, 2, \dots, m$, may vary in a continuum and the number of pattern classes belong to an integer set, i.e., the set $(0, 1, 2, \dots)$. Thus, the minimization of the total risk function with respect to the number of classes should only be carried over this integer set. This restriction prevents us from employing a straightforward variational method to minimize R as is done in (7). However, we can modify the method by simply using the following procedure. Recalling the general expression for the total expected risk function

$$R = E_X \left\{ \int_m F(x, m, \hat{m}) P(m/x) dm \right\} + E_X \left\{ \sum_{k=1}^{\hat{m}} I_k(x, \theta_1, \dots, \theta_{\hat{m}}) G_k(x, \theta_1, \dots, \theta_{\hat{m}}) \right\} \quad (8)$$

The first term corresponds to the loss due to the incorrect guesses of the number of pattern classes. Now, let both m and $\hat{m}$ assume the values of the integer set. Then the loss function $F(x, m, \hat{m})$ may be represented in a matrix form

$$[F(x, m, \hat{m})] = \begin{bmatrix} f_{00}(x) & f_{01}(x) & \dots & \dots & \dots \\ f_{10}(x) & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & f_{ij}(x) & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \quad (9)$$

where the component $f_{ij}(x)$ in the matrix is the loss due to the guess of i pattern classes while there are actually j classes present, $i = 0, 1, \dots$, and $j = 0, 1, \dots$. In general, $f_{ij}(x)$ is a function of the received pattern sample x and $[F(x, m, \hat{m})]$ is an infinite matrix.

We can rewrite the first part of (8) in term of the components in (9) as follows:

$$\begin{aligned} E_X \left\{ \int_m F(x, m, \hat{m}) P(m/x) dm \right\} &= E \left\{ F(x, m, \hat{m}) \right\} \\ &= E \left\{ F(x, 0, \hat{m}) \right\} P\{m=0\} + \dots + E \left\{ F(x, j, \hat{m}) \right\} P\{m=j\} + \dots \\ &= E \left\{ f_{\hat{m}0}(x) \right\} P\{m=0\} + \dots + E \left\{ f_{\hat{m}j}(x) \right\} P\{m=j\} + \dots \end{aligned} \quad (10)$$

Let $F_{\hat{m}}$ = a Hilbert space vector with components

$$E \left\{ f_{\hat{m}0}(x) \right\}, E \left\{ f_{\hat{m}1}(x) \right\}, \dots, E \left\{ f_{\hat{m}j}(x) \right\}, \dots$$

and P = a Hilbert space vector with components

$$P\{m=0\}, P\{m=1\}, \dots, P\{m=j\}, \dots$$

where $P\{m=j\}$ = the marginal probability that there are actually j pattern classes present. The total expected risk function becomes

$$R = F_{\hat{m}} \cdot P + E_X \left\{ \sum_{k=1}^{\hat{m}} I_k(x, \theta_1, \dots, \theta_{\hat{m}}) G_k(x, \theta_1, \dots, \theta_{\hat{m}}) \right\} \\ = F_{\hat{m}} \cdot P + R_2(\theta_1, \dots, \theta_{\hat{m}}) \quad (11)$$

The following procedure is proposed to minimize R in equation (11) or (8), depending on the type of knowledge about the distribution of m . First, the learning recognition system computes the probability vector

$$P = (P\{m=0\}, P\{m=1\}, \dots)$$

If (8) is used, the system computes $P(m/x)$ from the received sample x . Secondly, for each value of $\hat{m}$, the system finds an optimal set of values for θ_k , $\hat{\theta}_k$, by using the variational method [3, 7] and performing the following operations on $R_2(\theta_1, \dots, \theta_{\hat{m}})$.

$$\frac{\partial R_2}{\partial \theta_i} = E_X \left\{ \sum_{k=1}^{\hat{m}} I_k(x, \theta_1, \dots, \theta_{\hat{m}}) \frac{\partial G_k(x, \theta_1, \dots, \theta_{\hat{m}})}{\partial \theta_i} \right\} \\ + E_X \left\{ \sum_{k=1}^m \frac{\partial}{\partial \theta_i} I_k(x, \theta_1, \dots, \theta_{\hat{m}}) G_k(x, \theta_1, \dots, \theta_{\hat{m}}) \right\} \\ = 0 \quad i = 1, 2, \dots, \hat{m}. \quad (12)$$

The solution of equation (12) gives the optimal set of values for $\hat{\theta}_k$, $k = 1, 2, \dots, \hat{m}$, for each value of $\hat{m}$. Furthermore, according to (7), the gradient of the characteristic function $\frac{\partial}{\partial \theta_i} I_k(x, \theta_1, \dots, \theta_{\hat{m}})$ is a

generalized function on a surface --- a multi-dimensional delta function. Therefore the second term in (12) is zero for all x except for the points $x = x^0$ which belong to the boundaries B_{ij} between adjacent regions X_i and X_j . We now require that for $x = x^0$

$$G_i(x^0, \theta_1, \dots, \theta_{\hat{m}}) = G_j(x^0, \theta_1, \dots, \theta_{\hat{m}}) \quad (13)$$

This implies that we assign equal loss function for pattern samples which may be considered to come from either one of the two adjacent pattern classes (boundary samples). Then the condition in (12) reduces to

$$\frac{\partial R_2}{\partial \theta_i} = E_X \left\{ \sum_{k=1}^{\hat{m}} I_k(x, \theta_1, \dots, \theta_{\hat{m}}) \frac{\partial G_k(x, \theta_1, \dots, \theta_{\hat{m}})}{\partial \theta_i} \right\} \\ = 0 \quad i = 1, 2, \dots, \hat{m}. \quad (14)$$

The optimal set of estimates $\hat{\theta}_k$ is the solution of (14). The minimized risk function is then calculated by substituting $\hat{\theta}_k$ into equation (11) for θ_k . Note that the minimized risk function obtained is a function of $\hat{m}$, denoted by $R(\hat{m})$. The third step is to perform further minimization of $R(\hat{m})$ by the optimal choice of $\hat{m}$, denoted by $\hat{m}_{opt}$. That is, $\hat{m}_{opt} = \min_{\hat{m}} R(\hat{m})$.

Finally, it is necessary to determine the decision rules in order to

assign class membership to each and every pattern sample received. This is done by using the information of $\hat{m}_{opt}$ and $\hat{\theta}_k$, $k = 1, 2, \dots, \hat{m}_{opt}$, obtained above.

$$\begin{aligned} \text{Let } G_{ij}(x, \hat{\theta}_1, \dots, \hat{\theta}_{\hat{m}_{opt}}) \\ = G_i(x, \hat{\theta}_1, \dots, \hat{\theta}_{\hat{m}_{opt}}) - G_j(x, \hat{\theta}_1, \dots, \hat{\theta}_{\hat{m}_{opt}}) \end{aligned} \quad (15)$$

where $x^\circ \in B_{ij}$.
If, for all $j = 1, 2, \dots, \hat{m}_{opt}$, $j \neq i$

$$G_{ij}(x, \hat{\theta}_1, \dots, \hat{\theta}_{\hat{m}_{opt}}) < 0, \text{ then } x \in x_i \quad (16)$$

It has been shown in [3] that the decision rule expressed in (16) can be uniquely defined by the loss functions $G_k(x, \hat{\theta}_1, \dots, \hat{\theta}_{\hat{m}_{opt}})$. Thus, if we produce the parameters, $\hat{\theta}_1, \dots, \hat{\theta}_{\hat{m}_{opt}}$ and $\hat{m}_{opt}$ in a recursive fashion, the problem of simultaneous detection and estimation of pattern characteristics is solved. We shall discuss a recursive algorithm in the next section.

IV. RECURSIVE LEARNING ALGORITHM

Successive estimates for both $\hat{m}_{opt}$ and $\hat{\theta}_k$ may be obtained by applying the well-known stochastic approximation algorithm. Let the sequence of received pattern samples be denoted by $x(1), x(2), \dots, x(n), \dots$. $x(n)$ is the sample received at the n th stage of the learning recognition process. Let $\hat{\theta}_k(n)$ be the n th estimate of the parameter θ_k , $k = 1, 2, \dots, \hat{m}_{opt}(n)$, where $\hat{m}_{opt}(n)$ is the n th estimate of m . Then, according to (14), we have the following recursive formulas:

$$\begin{aligned} \hat{\theta}_k(n) = \hat{\theta}_k(n-1) - \gamma_k(n) \left\{ \sum_{k=1}^{\hat{m}_{opt}(n-1)} I_k[x(n), \hat{\theta}_1(n-1), \dots, \right. \\ \left. \hat{\theta}_{\hat{m}_{opt}(n-1)}(n-1)] x \frac{\partial}{\partial \theta_i} G_k[x(n), \hat{\theta}_1(n-1), \dots, \hat{\theta}_{\hat{m}_{opt}(n-1)}(n-1)] \right\} \\ k = 1, 2, \dots, \hat{m}_{opt}(n-1). \end{aligned} \quad (17)$$

and $I_k[x(n), \hat{\theta}_1(n-1), \dots, \hat{\theta}_{\hat{m}_{opt}(n-1)}(n-1)]$

$$= \begin{cases} 1 & \text{if } G_{kj}[x(n), \hat{\theta}_1(n-1), \dots, \hat{\theta}_{\hat{m}_{opt}(n-1)}(n-1)] < 0 \text{ for all } j \neq k \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Note that if $\hat{m}_{opt}(n-1)$ is not equal to $\hat{m}_{opt}(n-2)$, only $\hat{m}_{opt}(n-1)$ of the parameters $\hat{\theta}_k(n-1)$ are updated by using (17) and (18). The new $\hat{m}_{opt}(n)$ at the n th stage is obtained by substituting $\hat{\theta}_k(n)$ into equation (11), thus yielding $R(\hat{m})$ and the $\hat{m}$ which gives $\min R(\hat{m})$ is $\hat{m}_{opt}(n)$.

For the learning algorithm (17)-(18) to converge to the true parameters θ_k characterizing the pattern classes with probability one, the following conditions must be satisfied:

(1) The numerical sequence $\gamma_k(n)$ satisfies

$$\gamma_k(n) > 0, \quad \sum_{n=1}^{\infty} \gamma_k(n) = \infty \quad \text{and} \quad \sum_{n=1}^{\infty} \gamma_k^2(n) < \infty \quad (19)$$

(2) The loss function G_k must be differentiable with respect to the parameters θ_k .

It should be noted that algorithm (17)-(18) may also be written in a continuous fashion for the purpose of analog computation when it is more convenient. For example, if x becomes a continuous function of time t , then it is possible to write

$$\frac{d\hat{\theta}_k(t)}{dt} = -\gamma_k(t) \left\{ \sum_{k=1}^{\hat{m}_{\text{opt}}(t)} I_k[x(t), \hat{\theta}_1(t), \dots, \hat{\theta}_{\hat{m}_{\text{opt}}(t)}(t)]x + \frac{\partial}{\partial \hat{\theta}_1} G_k[x(t), \hat{\theta}_1(t), \dots, \hat{\theta}_{\hat{m}_{\text{opt}}(t)}(t)] \right\} \quad (20)$$

The conditions for $\gamma_k(t)$ become

$$\gamma_k(t) > 0, \quad \int_0^{\infty} \gamma_k(t) dt = \infty \quad \text{and} \quad \int_0^{\infty} \gamma_k^2(t) dt < \infty \quad (21)$$

Let us review and summarize the recursive algorithm for simultaneous detection and estimation in a self-learning recognition system:

- (1) The system assigns loss functions $F(x, m, \hat{m})$ and $G_k(x, \theta_1, \dots, \theta_k)$ and makes initial guesses for $\hat{m}$ and $\hat{\theta}_k$, $\hat{m}_{\text{opt}}(0)$ and $\hat{\theta}_k(0)$, respectively. Either the a priori knowledge of $P\{m=j\}$ or the a posteriori probability $P(m/x)$ is assumed to be known (or can be approximated).
- (2) From the initial guesses and the first received pattern sample $x(1)$, the system computes $\hat{m}_{\text{opt}}(1)$ and $\hat{\theta}_k(1)$ using (17). If $P(m/x)$ is known, then the minimization process should be carried directly over equation (8) and (17) should be modified accordingly.
- (3) The system assigns class membership for $x(1)$ according to the decision rule established in (16).
- (4) Step (2) and (3) are repeated when $x(2)$, $x(3)$, etc. are successively received.

V. CONCLUSIONS

We have discussed a recursive technique for simultaneously performing two tasks in a self-learning recognition system, namely, the detection of how many pattern classes are present and the estimation of pattern parameters for different pattern classes. The optimal classification of pattern samples then becomes a direct consequence of these two tasks. In the proposed technique, the simultaneous detection and estimation involve the required knowledge of either the a priori probability of the number of classes, m , or the a posteriori probability of m , given the pattern sample x . In addition, loss functions due to incorrect classification must satisfy certain condition in order to achieve minimum risk. No attempt has been made to determine specific forms of the learning algorithm when various loss functions are chosen.

Several topics may be suggested as problems for further study:

- (1) Analysis of methods to calculate recursively the a posteriori probabilities for the number of pattern classes present as samples are acquired.
- (2) Investigation of specific forms of the learning algorithm resulting from various loss functions.
- (3) Generalization of this discussion to the situation where m may assume values in a continuum. This situation arises when no definite class membership can be assigned to each and every pattern sample in a recognition process.

REFERENCES

1. Nilsson, N. J., "On optimum range resolution of radar signals in noise," IRE Transaction on Information Theory, IT-7, 1961, pp. 245-253.
2. Ball, G. H., "Data analysis in the social sciences: what about the details?" Proc. FJCC, December 1965, pp. 533-559.
3. Tsytkin, Ya. Z. and G. K. Kelmans, "Recursive algorithms of self-learning," Engineering Cybernetics, no. 5, 1967, pp. 70-80.
4. Middleton, D. and R. Esposito, "Simultaneous optimal detection and estimation of signals in noise," IEEE Transaction on Information Theory, IT-14, no. 3, May 1968, pp. 434-444.
5. Latysh, V. G., "Separating group of unresolved signals from noise using random point theory," Engineering Cybernetics, no. 2, 1967, pp. 42-51.
6. Balshakov, I. A., "Definition of the intragroup structure of one and many groups of a random membe of signals," Engineering Cybernetics, no. 3, 1965.
7. Tsytkin, Ya. Z., "Self-learning -- what is it?" to be published in the IEEE Transaction on Automatic Control.
8. Chien, Y. T. and K. S. Fu, "On Bayesian learning and stochastic approximation," IEEE Transaction on Systems Science and Cybernetics, Vol. SSC-3, no. 1, January 1967, pp.28-38.

THE HIGHER PROBABILITY NEIGHBORHOOD DECISION RULE

GORDON CHANG, S. J. DWYER III, JOHN J. KOMO
University of Missouri - Columbia, Columbia, Missouri

The discrimination problem in its simplest form consists of two states of nature, θ_i , $i = 1, 2$. A measurement vector x on a d -dimensional vector space X is observed and a decision is then made as to which state the x is from. Let $f_i(\theta_i, x)$, $i=1, 2$, be the joint distribution of θ_i and x , where the form of $f_i(\theta_i, x)$ is unknown. The particular distribution-free decision rule selected for investigation is the higher probability neighborhood (HPN) decision rule. Assume that a set of n training samples $\{\theta_i^1, x_i^1\}_{i=1}^n$ in the ϵ -neighborhood of x , $N(x, \epsilon)$, has been observed. The HPN decision rule states that:

"For a given state-measurement pair (θ_i, x) , x is assigned to the state θ_i having the relative higher occurrence on the ϵ -neighborhood of x ." In practice, the HPN decision rule assigns x to the θ_i from which the majority of the x^1 's of the training sample come. The "majority" used here is not the arithmetic "majority" but rather statistical "majority".

Define, for $x \in N(x, \epsilon)$

$$\begin{aligned} \delta(x^1) &= 1, & \text{if } x^1 \text{ is from } \theta_1 \\ \delta(x^1) &= 0, & \text{if } x^1 \text{ is from } \theta_2 \end{aligned}$$

Letting μ be a σ -finite measure defined on X , then

$$p = E \{ \delta(x^1) \} = \frac{\int f_1(\theta_1, x) d\mu}{\int [f_1(\theta_1, x) + f_2(\theta_2, x)] d\mu},$$

and $\hat{p} = \frac{1}{n} \sum_{i=1}^n \delta(x^i)$ is an unbiased estimator of p which converges almost surely to p .

The HPN decision rule essentially is to test hypothesis $H: p = \frac{1}{2}$ against $H_A: p \neq \frac{1}{2}$. The HPN decision rule will assign x to the state θ_1 if the test is significant and $\hat{p} > \frac{1}{2}$; otherwise assign x to the state θ_2 . By using the HPN decision rule, we may, therefore, give a confidence statement to the decision we made. In case the test is not significant, but $\hat{p}$ is greater than $\frac{1}{2}$, we still assign x to the state θ_1 , and to the state θ_2 if $\hat{p}$ is less than $\frac{1}{2}$; no confidence statement can be made to our decision. The error of misclassification by using the HPN decision rule is less than $(1 - \hat{p} + \lambda)$, where λ is a positive number whose value can be made as small as desired.

Given an observed value of p , the probability of error of the HPN decision rule can be determined. Let $\gamma(\hat{p})$ be the conditional probability of misclassification of the unidentified measurement x , then for given α, λ , however small, there exists an $N = 1/(4\lambda^2\alpha)$ by using Tchebichev's inequality, such that for $n \geq N$, n is the size of the training sample,

$$P[\gamma(\hat{p}) \leq 1 - \hat{p} + \lambda] \geq 1 - \alpha,$$

where α, λ are such that $P[|\hat{p} - p| < \lambda] \geq 1 - \alpha$. We can say then the $100(1-\alpha)\%$ confidence interval for p is $(p-\lambda, p+\lambda)$.

Since $\hat{p}$, without loss generality is greater than $\frac{1}{2}$, and λ is sufficiently small, the probability of error of misclassification of x given $\hat{p}$ is always less than $\frac{1}{2}$, and it will tend to zero as $\hat{p}$ tends to one as we may see from above.

One is often faced with the knowledge that the training sample set $\{\theta_i^1, x_i^1\}_{i=1}^n$ is in error. Assume such errors are independent and identically

distributed, and with probability $1 - \beta$ that each x^i would be misclassified. Then it is readily shown that $\hat{p}$ may be adjusted as,

$$\hat{p}_{\text{adj}} = \frac{\sum_{i=1}^n \delta(x^i) + n(\beta-1)}{n(2\beta-1)}$$

In the HPN decision rule, we may also use weighted estimate of p . A suitable weighting factor may relax the restriction on the ϵ -neighborhood of x , and also effect the required size of training sample. For example, the reciprocal of the distance from x , or the square of the reciprocal distance from x , may be used as weighting factor. The suitable choice of the weighting factor depends on the nature of the problem.

The extension of the HPN decision rule from two-states to m states is straightforward. Let $\{\theta^i, x^i\}_{i=1}^n$ where $n > m$, be the training samples in the ϵ -neighborhood, and define the function $\delta_i(x^j)$ as follows

$$\delta(x^j) = \begin{cases} 1, & \text{if } x^j \text{ is from } \theta_1 \\ 0, & \text{otherwise,} \end{cases}$$

where $i=1,2, \dots, m, j=1,2, \dots, n$. Also, define

$$p_i = \frac{\int f_i(\theta_i, x) d\mu}{\int \left[\sum_{i=1}^m f_i(\theta_i, x) \right] d\mu} \quad N(x, \epsilon), \quad \hat{p} = \frac{1}{n} \sum_{j=1}^n \delta_i(x^j), \quad i=1,2, \dots, m.$$

Clearly $E\hat{p}_i = p_i$, and $\hat{p}_i$ converges to p almost surely. The HPN decision rule assigns x to the state θ_i if

$$\hat{p}_i = \text{Max}(\hat{p}_1, \hat{p}_2, \dots, \hat{p}_m), \text{ with } \sum_{i=1}^m \hat{p}_i = 1.$$

The probability of error of misclassification of x given $\hat{p}_i$ is

$$P[Y(\hat{p}_i) \leq 1 - \hat{p}_i + \lambda] \geq 1 - \alpha,$$

where α, λ are such that $P[|\hat{p}_i - p_i| < \lambda] \geq 1 - \alpha$, and the $100(1-\alpha)\%$ confidence interval for p_i is $(\hat{p}_i - \lambda, \hat{p}_i + \lambda)$. Since $\hat{p}_i$ is not necessarily greater than $\frac{1}{2}$, we cannot conclude that the probability of error of misclassification of x is less than $\frac{1}{2}$ as we did in the two-state situation.

Under the assumption that for a given ϵ -neighborhood of x , the size of the training samples may increase indefinitely, there is a subsequence of the training samples which converges to x almost surely. In the two-state discrimination problem it is readily shown that the HPN decision rule is equivalent to the optimum decision rule. The optimum decision rule is

"Given $x \in N(x, \epsilon)$, decide $\hat{\theta} = \theta_i$ if and only if

$$p[\theta_i | x \in N(x, \epsilon)] > p[\theta_j | x \in N(x, \epsilon)], \quad i \neq j, \quad i, j=1,2."$$

which is the same as the HPN decision rule that "if $\hat{p}$ is greater than $\frac{1}{2}$ assign x to state θ_i , otherwise to state θ_2 ."

JOHN J. METZNER AND KENNETH C. MORGAN
New York University, Bronx, New York 10453

ABSTRACT

A variable-data-rate procedure employing a fixed word duration is described. The procedure is shown to be compatible with long-code, fail-safe communication. It also permits independent rate variations in the opposing directions of a two-way communication system - a procedure which results in a substantial improvement in communication efficiency.

The advantages of independent rate variation are illustrated. For an assumption of identical but independent Rayleigh-distributed slow fading over the two channels, the independent rate-variation procedure shows a 3:1 advantage in average rate over a fixed, ideal, cumulative-decision system and a 2:1 advantage over a variable-data-rate system in which the rate is the same in both directions. Another model shows an even greater advantage.

I. INTRODUCTION

It is clear that many real communication channels do not have statistical properties which are stationary in time or highly predictable in advance. The information capacity of such a channel may be thought of as varying in time. For maximum transmission efficiency, then, the method of transmission (especially the rate of transmission) should be varied according to channel conditions.

Section II shows that varying-data-rate procedures are compatible with long-code, fail-safe procedures.^{1,2} Discussion is concentrated on one particularly attractive method (the variable-information-content method) which permits changing the rate in one direction of a two-way communication system independently of the rate in the other direction in a relatively straightforward manner. Section III provides some quantitative measures of the inherent advantage of independent rate changes in the two directions.

II. A VARIABLE WORD-CONTENT PROCEDURE

Propagation and encoding-decoding delays require that a number of words be sent out before it is possible to act on a confirmation or repeat request for the first word (this number has been called the transmission cycle parameter J).³ If the data rate varies and the word information content is fixed, the time duration of a word changes. Therefore, the number of words which are sent out before a repeat request for the first can be acted on may change. Then again, rather than change the cycle parameter J with each data rate change, it is possible to maintain a constant J corresponding to that required for the highest rate by deliberately holding up action on repeat requests when the data rate is low. The logical rules would not then be altered, but simply operate at a different basic rate of the data rate were changed.

An alternative to the above is to vary the information content of a word according to the data rate, maintaining a fixed word duration. That is, more information would be "packed" into a word at high rates than at low rates. This packing could be done by means of a multi-level (instead of binary) mode for each digit, or by increasing the digit rate.

*This research was sponsored by the Air Force Cambridge Research Laboratories, Office of Aerospace Research, under Contract AF19(628)-4321.

An attractive feature of the variable-word-content procedure is the ease with which the transmitted rate can be changed in one direction without changing it concurrently in the other direction. A weakness of those two-way feedback systems which are restricted to equal or proportional rates in both directions is that the effective rate is controlled by the weaker of the two links. If, for example, the two links fade independently of each other, an appreciable acceptance rate occurs only in intervals when both channels are good. However, if the two rates are changed independently as in the variable-content procedure, the weaker channel can send at a slower information rate. This increases the probability of getting a confirmation signal through the weaker channel. In this way, the stronger channel is not held up.

In order to retain the fail-safe and error-free features of the basic feedback communication system, any modification must retain the two essential characteristics of the basic system. The first is that when a word is accepted the probability of an error must be negligible. This is readily achieved, as before, by utilizing a sufficient number of check digits relative to the amount of error correction attempted. The second is that procedural rules, and associated logic circuitry, must permit the message to be reconstructed unambiguously from the various word acceptances. The following argument shows that this is achieved in the present case.

Two word-selection procedures - the circulating memory procedure and the interlacing procedure - have been discussed previously.^{2,4} Of the two, the interlacing procedure is better suited to handle variable-content words, for reasons explained below.

The circulating and interlacing procedures basically differ only in the manner of selection of the next new word. Both utilize a temporary storage "ring" capable of holding J words (see Figure 1). This ring may be thought of as a shift register which shifts the length of one word per word transmission. In the circulating procedure (Figure 1a) the switch is placed in position R when a repeat is called for, in which case the contents of K_J (the same word sent J word units ago) is transmitted and also is shifted into K_1 . Simultaneously, the old contents of K_i go into K_{i+1} , $i=1$ to $J-1$. If the previous word was confirmed, so that a new word may be sent, the switch is placed in position N , and the register is then shifted, sending the new word into K_1 , and also transmitting it. The confirmed word K_J is shifted out and discarded. Note that new words enter the register in the order they appear in the original message, but they are not necessarily accepted at the other station in that order. A labeling scheme prevents ambiguity. (see reference 4, sections III.A.1, III.A.3). This consists of ℓ labeling digits per word, together with the rule that the transmitter will not send out any word advanced by $2^{\ell}J$ (Method A) or $2^{\ell}J-1$ (Method B) from the oldest word still circulating in transmitter memory.

In the interlacing procedure the encoded message is divided into J sets in the following manner: set A_i contains the i^{th} , $(J+i)^{\text{th}}$, $(2J+i)^{\text{th}}$, ... words. A J -position cyclic stepping switch steps in synchronism with the shifts of the shift register, ensuring that the $(kJ+i)^{\text{th}}$ transmission is always a word from the set A_i . When a word is confirmed, it will always be replaced in the shift register by a word from the same set. Transmission of each of the J sets proceeds, in effect, independently of the others, but interlaced with them as described.

Consider now the effect of a change in the amount of word content. In the circulating case the record of accepted words usually contains gaps. When the word content is changed these gaps are no longer word units. To prevent confusion a rather elaborate system of rules and labeling is needed.

In the interlacing procedure a change in rate is more easily handled. Let each word unit contain two labels. One specifies from which set A_i is taken (this requires at least $\log_2 J$ bits). The second is a single label digit which alternates between 0 and 1 for successive word units in the set.

Figure 2 shows how the interlacing method can be implemented at the transmitter. When sending at the lowest rate only Section 1 of the register is used. At the highest rate (assumed to be m times the lowest) all m sections are used. At fixed rate the operations are just as before (Figure 1), except that it is specified here that the encoding is done at the transmitter. By a modified procedure the encoding could be done prior to entry into the register. When a rate change is to be made, a slightly different mode is entered for the first J transmissions at the new rate (see Figure 2). The input switch is kept at N , but nothing is read in until after the register has shifted. If a confirmation is received, the output switch is placed at R and the register is shifted with the contents of K_j discarded. Next, a word at the new rate is entered into K_1 without shifting the register and is also encoded and transmitted. The alternating label digit is opposite to that contained in the contents of the word just discarded. If a confirmation is not received, the output switch is set at "return", and the contents of K_j are fed back to the appropriate set in the proper order as the register is shifted. Then, a word at the new rate is taken from this same set (it will be part of or include the word unit just put back) and entered into K_1 , encoded and transmitted. In this case the alternate label digit is caused to be the same as in the contents of the word just returned from K_j . After J transmissions the system returns to the fixed-rate mode, with the register output switch always set at R .

At the other station the receiver may or may not become aware of the change in rate until a word is decoded at the new rate. However, the fail-safe property permits attempts to decode in more than one mode. If the newly received word has the same label digit as the word previously accepted from the same set, that previously accepted word is discarded, for either the newly accepted word will include the previously accepted word as its first part (if the new rate is higher) or the previously accepted word will constitute the contents of a number of successive new words of that set (if the new rate is lower).

III. AN ILLUSTRATION OF THE EFFECTIVENESS OF INDEPENDENT RATE VARIATIONS

Assume a two-way, coded, feedback communication system in which message information is being sent in both directions. The coding protection can be made sufficient that we may neglect errors in accepted received words. The remaining consideration is to obtain the highest possible acceptance rate compatible with this virtually error-free operation. In order to advance to a new word, not only must a given transmitted word be accepted at the other station, but the returned word which contains its confirmation must also be accepted at the original station. Now if conditions on the two links are varying slowly, but independently of each other, the short-term rate will always be limited by whichever of the links happens to be weaker at a given time. If, however, the rate of the weaker link were reduced unilaterally to a value which yields a high rate of acceptance, it would not hold up the strong link. At other times, when the role of strong and weak links were reversed, a high rate would be possible in the then strong direction. In effect, the weaker link would always be able to help the stronger one, and the result should be a higher average rate of accepted information than could be attained if both rates had to be in fixed proportion.

Suppose that a variable-data-rate scheme is restricted to a finite set of rates $\{r_0, \dots, r_N\}$ where i is an integer, $1 \leq i \leq N$. This may be a practical requirement. If one link is error-free, such a variable-data-rate scheme is no more effective than an idealized cumulative decision scheme operating at the highest rate r_0 . By an idealized cumulative decision scheme we mean that statistics for a cumulation of n repeats of a word at rate r_0 are the same as for a single transmission of a word at the rate r_0/n . In the variable rate

case, sending a rate r_0/n corresponds to sending some fixed number n of repeats or some multiple thereof, and this will often be more than necessary. In the cumulative case, exactly as many repeats are supplied as are needed. However, this advantage of the cumulative scheme is probably minor relative to departures from the ideal of both techniques in practical systems.

Where both channels are imperfect, however, the variable data rate approach may show a decided advantage over cumulative techniques for combating long-term channel variations,* even if the two rates cannot be varied independently. For example, suppose that with cumulation at rate r_0 it takes an average of n transmissions in either direction to get a word through to the other station. It has been shown in Appendix B of reference 1 that the normalized transmission time (ratio of total number of words transmitted to the number of new words transmitted) is given by

$$T_t = T_{t1} + T_{t2} - 1 \quad (1)$$

where T_{t1} would be the normalized transmission time if channel 1 were noiseless. Thus, for our assumption, we have $T_{t1} = T_{t2} = n$. Applying (1),

$$R_c = \frac{r_0}{2n-1} \quad (2)$$

If we sent at the rate $\frac{r_0 - \epsilon}{n}$ in both directions where ϵ is a small positive number, we would accept almost everything (if long code words are assumed the actual fraction of errors is nearly always close to the average), so that $T_{t1} = T_{t2} \approx 1$. Thus, the rate for the variable-data-rate case is

$$R_v \approx \frac{r_0}{n} \quad (3)$$

Comparison of (2) and (3) shows $R_v \approx 2R_c$ for large n .

When independent rate variations are permitted, still further improvements are possible. As a mathematical illustration, assume the following:

(1) Channel conditions on the two links are identically, but independently, distributed. The probability distribution of the amplitude signal-to-noise ratio is Rayleigh.

(2) Variations are slow, so that it always is possible to be at the proper rate.

(3) The links are both disturbed by white gaussian noise.

The following three systems are to be compared:

(1) An ideal fixed-rate cumulative decision feedback system operating at a very high basic rate.

(2) A variable-data-rate system with equal rates in both directions.

(3) A variable-data-rate system permitting independent rate variations.

Let ρ_1 be the signal-to-noise power ratio in direction (channel) one, and ρ_2 the similar ratio in direction (channel) two at digit rate r_0 . Also, let $p(\cdot)$ be a probability density function.

Imagine some very high basic rate r_0 , which we may eventually let approach infinity. For long codes, the average number of errors usually is close, percentagewise, to the actual number of errors. Thus, there is some sharp threshold signal-to-noise ratio ρ_c . If $\rho > \rho_c$, the word can nearly always be decoded. If $\rho < \rho_c$, it rarely can be decoded. Since we have white gaussian noise, stretching the signal by a factor $k > 1$ raises the effective power snr to $k\rho$. Thus, in an ideal cumulative system (where using n repeats

*Note that cumulative decision techniques are unquestionably preferable for combatting short-term channel fluctuations too rapid to be followed by varying the data rate. There is nothing, however, which prevents the concurrent use of both cumulative and variable-data-rate techniques in the same system.

is as effective as stretching by n) the number of cumulations usually is approximately the smallest integer exceeding ρ_c/ρ . Ignoring the restriction to integral values (since r_0 will be assumed indefinitely large anyway, the number of cumulations required becomes large, and the restriction to integers is a good approximation), we can write, for the cumulative case,

$$T_{t1}(\rho_1) \doteq \rho_c/\rho_1 \text{ if } \rho_1 < \rho_c \quad . \quad (4)$$

Cumulative-Decision System Acceptance Rate. For fixed ρ_1, ρ_2 ,

$$R_c = \frac{r_0}{T_t(\rho_1, \rho_2)} \quad . \quad (5)$$

Thus,

$$\bar{R}_c = r_0 \int_0^\infty \int_0^\infty \frac{p(\rho_1)p(\rho_2)}{T_t(\rho_1, \rho_2)} d\rho_1 d\rho_2 \quad , \quad (6)$$

under our assumption of ρ_1, ρ_2 independent and slowly-varying.

From (1),

$$\bar{R}_c = r_0 \int_0^\infty \int_0^\infty \frac{p(\rho_1)p(\rho_2)}{T_{t1}(\rho_1) + T_{t2}(\rho_2) - 1} d\rho_1 d\rho_2 \quad . \quad (7)$$

For $\rho_1, \rho_2 \ll \rho_c$,

$$T_{t1}(\rho_1) + T_{t2}(\rho_2) - 1 \doteq \rho_c/\rho_1 + \rho_c/\rho_2 \quad . \quad (8)$$

If r_0 is sufficiently large, $p(\rho_1) \doteq 0$ and $p(\rho_2) \doteq 0$ for values of ρ_1, ρ_2 sufficiently large that (8) is no longer a good approximation. Thus,

$$R_c \doteq \frac{r_0}{\rho_c} \int_0^\infty \int_0^\infty \frac{\rho_1 \rho_2}{\rho_1 + \rho_2} p(\rho_1)p(\rho_2) d\rho_1 d\rho_2 \quad . \quad (9)$$

Variable-but-Equal Rate System. In this case we assume that the rate is chosen just high enough that nearly all words are accepted. Thus,

$$T_t(\rho_1, \rho_2) = 1 + \delta \quad ,$$

where ρ is a small positive number which will be neglected.

If $\rho_2 > \rho_1$, we choose

$$r_1 = r_2 = \frac{r_0 \rho_1}{\rho_c} - \epsilon \quad , \quad (10)$$

where ϵ is a small (relative to r_1, r_2) positive number which we choose to ignore. For sufficiently long codes, δ and ϵ can simultaneously be made smaller than any preassigned positive number. In general,

$$r_1 = r_2 \doteq \frac{r_0}{\rho_c} \text{Min}\{\rho_1, \rho_2\} \quad . \quad (11)$$

Given the probability distributions $p(\rho_1), p(\rho_2)$, the probability

distribution $p(\rho_{\min})$ of the minimum of ρ_1 and ρ_2 can be determined. Then,

$$\bar{r}_2 = \bar{r}_1 = \int_0^{\infty} \frac{r_0}{\rho_c} \rho_{\min} p(\rho_{\min}) d\rho_{\min} = \frac{r_0}{\rho_c} \bar{\rho}_{\min} \quad (12)$$

Let $\bar{R}$ be the average acceptance rate in either direction. Since $T_t = 1$,

$$\bar{R}_{ev} \triangleq \bar{r}_1 = \frac{r_0}{\rho_c} \bar{\rho}_{\min} \quad (13)$$

Independent Rate Variations. Assume that for a particular pair of signal-to-noise ratios ρ_1, ρ_2 the rates r_1, r_2 are such that

$$r_1 = \frac{r_0 \rho_1}{\rho_c} - \epsilon, \quad r_2 = \frac{r_0 \rho_2}{\rho_c} - \epsilon, \quad (14)$$

where ϵ is again a negligibly small positive number. Thus, the rate is being reduced just sufficiently that a word is nearly always accepted. Let $R_1(\rho_1, \rho_2)$ be the acceptance rate over channel 1. Thus,

$$R_1(\rho_1, \rho_2) \triangleq r_1 \quad (15)$$

Let $\bar{R}_V$ be the average rate with independent variations. Thus,

$$\bar{R}_V = \bar{R}_1 = \bar{R}_2 = \bar{r}_1 = \bar{r}_2 = \int_0^{\infty} r_1(\rho_1) p(\rho_1) d\rho_1 \quad (16)$$

From (14),

$$\bar{R}_V = \frac{r_0}{\rho_c} \int_0^{\infty} \rho_1 p(\rho_1) d\rho_1 = \frac{r_0 \bar{\rho}_1}{\rho_c} \quad (17)$$

Comparisons. Since $\rho_1 > \frac{\rho_1 \rho_2}{\rho_1 + \rho_2}$, it is readily seen by a comparison of (9) and (17) that $\bar{R}_V > \bar{R}_c$. Also, since $\rho_1 \geq \rho_{\min}$, comparison of (13) and (17) shows that $\bar{R}_V > \bar{R}_{ev}$. Thus, the independent rate method yields the highest rate of the three systems. For a more quantitative comparison we shall consider two examples.

Example 1: Rayleigh Distribution of Signal-to-Noise Amplitude Ratio. Assume the amplitude probability distribution

$$q(x) = \frac{2x}{R} e^{-x^2/R} \quad x \geq 0; \quad q(x) = 0 \quad x < 0 \quad (18)$$

In terms of the power signal-to-noise ratio, $\rho = x^2$, a transformation of (18) yields

$$p(\rho) = \frac{1}{R} e^{-\rho/R} \quad (19)$$

Substituting (19) in (9),

$$\bar{R}_c = \frac{r_0}{\rho_c} \int_0^{\infty} \int_0^{\infty} \left\{ \frac{\rho_1 \rho_2}{\rho_1 + \rho_2} \right\} \frac{e^{-\rho_1}}{R} \left(\frac{\rho_1 + \rho_2}{R} \right) d\rho_1 d\rho_2 \quad (20)$$

Substituting $\rho_1 = r \cos \theta$, $\rho_2 = r \sin \theta$, and using integral formulas 3.667, no. 7 and 8.380 of reference 5, (20) eventually reduces to

$$\bar{R}_C = \frac{R r_0}{3 \rho_c} \quad (21)$$

For the independent-rate-variations case,

$$\bar{R}_V = \frac{r_0}{\rho_c} \int_0^{\infty} \frac{\rho_1}{R} e^{-\frac{\rho_1}{R}} d\rho_1 = \frac{r_0 R}{\rho_c} \quad (22)$$

Thus,

$$\bar{R}_V = 3 \bar{R}_C \quad (23)$$

For the equal-rate-variation case the probability density of $\rho_{\min}$ can be shown to be

$$p(\rho_{\min}) = \frac{d}{d\rho} \{ \text{Prob.} [\rho_{\min} < \rho] \} = \frac{2}{R} e^{-\frac{2\rho}{R}} \quad (24)$$

Substituting (24) in (12) and integrating,

$$\bar{R}_{ev} = \frac{r_0 R}{2 \rho_c} \quad (25)$$

Thus, we have $\bar{R}_V = 2 \bar{R}_{ev} = 3 \bar{R}_C$. (26)

Example 2: Two Levels of Channel Condition. Suppose that on each channel the signal-to-noise power ratio is, independently of the other channel, ρ for 90% of the time, and is 100ρ for 10% of the time. Once in one state, the channel remains there for a long time, so that changes are relatively infrequent. Since the highest signal-to-noise ratio is known to be 100ρ , allow the cumulative system the benefit of a basic rate r_0 which is the highest for which nearly all words would be accepted when the signal-to-noise ratio is 100ρ on both channels. Any higher rate would diminish average performance. For this illustration the independent-rate-variation procedure can be shown to have about 5 times the average rate of the identical-rate procedure, and 6 times the average rate of the idealized cumulative-decision system.

IV. CONCLUSIONS

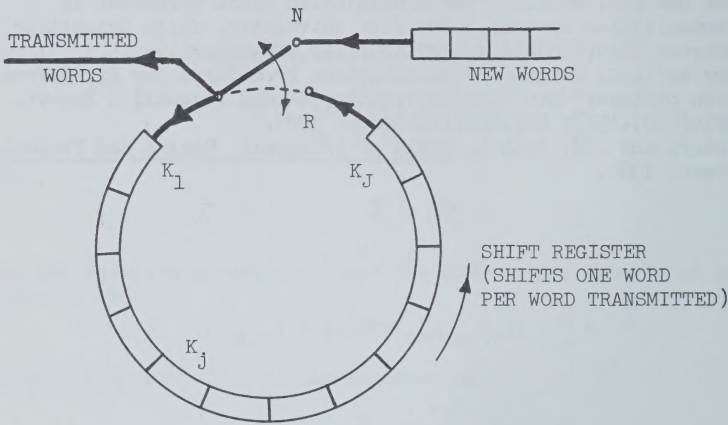
The fixed-word-duration, variable-data-rate procedure has been demonstrated to be compatible with long-code, fail-safe communication. This procedure provides one simple solution to the problem of varying the rate independently in the two directions. By avoiding the situation in which the rate of information acceptance is controlled by the weaker link, a substantial improvement in overall communication efficiency can be achieved.

Ideally, it would be desirable to always be sending at the most efficient rates. Practically, this is impossible because of the time required to measure and respond to variations in the condition of the channels. It seems clear that variable-data-rate techniques should be used to adapt to intermediate and long-term fluctuations in channel conditions. Cumulative decision techniques make it possible to remain efficient during short-term fading or burst-type disturbances.

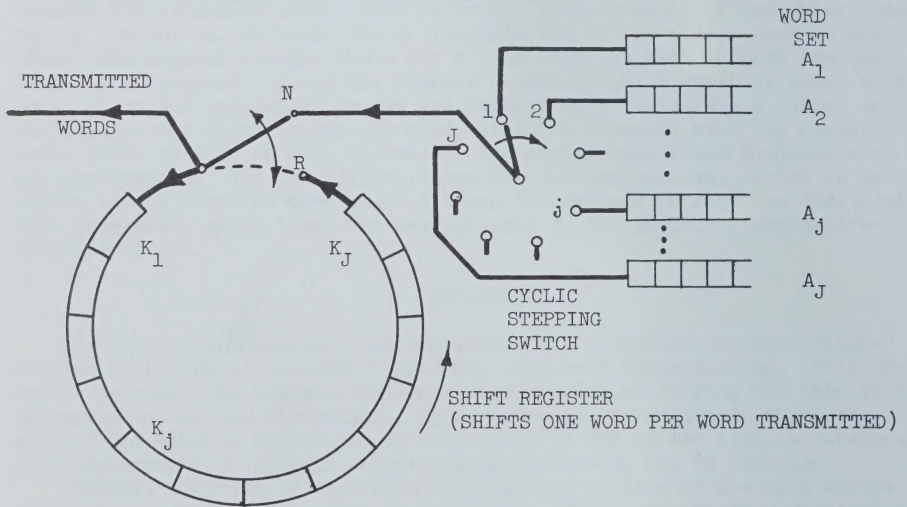
REFERENCES

1. J.J. Metzner and K.C. Morgan, "Cumulative Decision Techniques for Error-Free Communication Systems, New York University, Fourteenth Scientific

- Report, Contract AF19(604)-6168, UNCLASSIFIED. Sec. F., July 15, 1963.
2. J.J. Metzner and K.C. Morgan, "Reliable Fail-Safe Binary Communication," New York University, Second Scientific Report, Contract AF19(604)-6168, UNCLASSIFIED, July 10, 1960.
 3. J.J. Metzner and K.C. Morgan, "The Transmission Cycle Parameter in Feedback Communication Systems," New York University, Third Scientific Report, Contract AF19(628)-4321, UNCLASSIFIED, November 1965.
 4. J.J. Metzner and K.C. Morgan, "Word-Selection Procedures for Error-Free Communication Systems," New York University, Second Scientific Report, Contract AF19(628)-4321, UNCLASSIFIED, June 1965.
 5. I.S. Gradshteyn and J.M. Ryzhik, Table of Integrals, Series and Products, Academic Press, 1965.



(a) CIRCULATING TRANSMISSION PROCEDURE



(b) INTERLACING TRANSMISSION PROCEDURE

FIGURE 1. USE OF THE STORAGE RING IN THE CIRCULATING AND INTERLACING TRANSMISSION PROCEDURES.

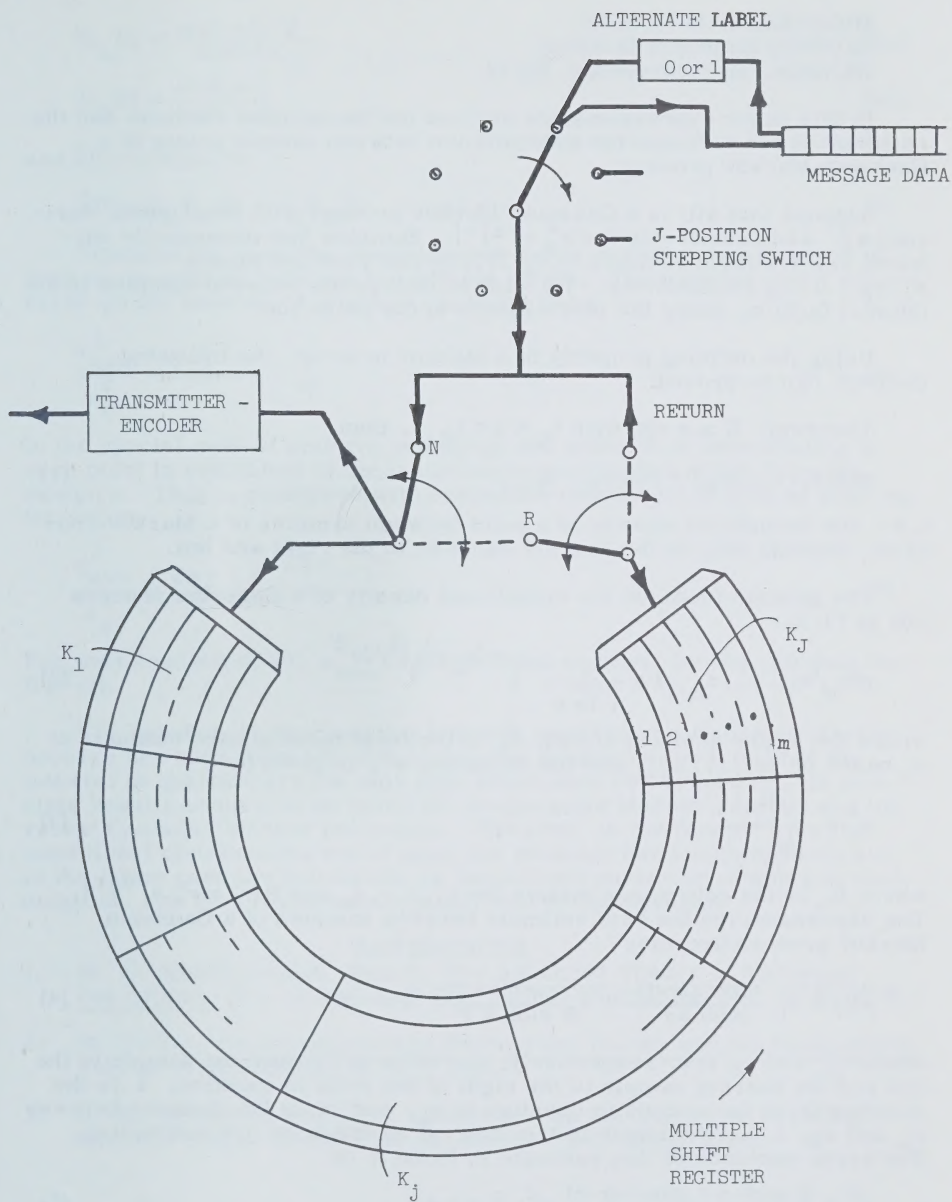


FIGURE 2. THE INTERLACE TRANSMISSION PROCEDURE IN A
FIXED WORD-DURATION, VARIABLE DATA RATE SYSTEM

INTERPOLATION OF A GAUSSIAN MARKOV PROCESS

MICHAEL A. SCHUMER
Raytheon Research Division
Waltham, Massachusetts 02154

In this paper expressions are derived for the optimal estimate and the resultant error variance for interpolation between sample points of a Gaussian Markov process.

Suppose that $x(t)$ is a Gaussian Markov process with zero mean, variance σ_x^2 , and autocorrelation $\sigma_x^2 e^{-a|\tau|}$. Samples (not necessarily uniform) are available at times $t_1, t_2, \dots, t_N$ and have the values $x_1, x_2, \dots, x_N$ respectively. We wish to interpolate between samples in an optimal fashion, using the mean square error criterion.

Using the defining property of a Markov process, the following theorem can be proved:

Theorem: If $x = x(t)$ with $t_K < t < t_{K+1}$, then

$$p(x/x_1, \dots, x_N) = p(x/x_K, x_{K+1}) \quad (1)$$

i. e., the probability density of a point between samples of a Markov process, depends only on the nearest samples to the right and left.

The general form for the conditional density of a Gaussian process $z(t)$ is [1, 2]:

$$p(z_n/z_1, \dots, z_{n-1}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z_n - \hat{z}_n)^2/2\sigma^2} \quad (2)$$

where the z_i are samples of $z(t)$, $\hat{z}_n$ is the least mean square estimate of z_n based on $z_1, \dots, z_{n-1}$ and the variance, σ^2 , is given by

$$\sigma^2 = \frac{|R_n|}{|R_{n-1}|} \quad (3)$$

where R_n is the covariance matrix for $z_1, \dots, z_n$ and $R(\tau) = \sigma_x^2 e^{-a|\tau|}$. The expression for the best estimate between samples of a Gaussian Markov process becomes

$$\hat{x}(t) = x_L \frac{\sinh a(\tau-t)}{\sinh a\tau} + x_R \frac{\sinh at}{\sinh a\tau}; \quad 0 \leq t \leq \tau \quad (4)$$

where x_L and x_R are, respectively, the value of the nearest sample to the left and the nearest sample to the right of the point in question, t is the distance from the sample in question to x_L , and τ is the distance between x_L and x_R , i. e., the length of the interval on which we are estimating. The error variance of this estimate is found to be

$$\sigma^2 = \frac{2 \sinh at \sinh a(\tau-t)}{\sinh a\tau} \sigma_x^2; \quad 0 \leq t \leq \tau \quad (5)$$

For $\tau \rightarrow \infty$, only one point is used in the estimate, and Eqs. (4) and (5) reduce to the well known extrapolation formulas of Uhlenbeck and Ornstein [3].

Equation (4) expresses $\hat{x}(t)$ as the sum of an estimate due to x_L and an estimate due to x_R . The normalized estimates can be written as

$$K_L(t) = \frac{\sinh a(\tau-t)}{\sinh a\tau} \quad (6a)$$

$$K_R(t) = \frac{\sinh at}{\sinh a\tau} \quad (6b)$$

and $\hat{x}(t)$ is equal to

$$\hat{x}(t) = x_L K_L(t) + x_R K_R(t) \quad 0 \leq t \leq \tau \quad (7)$$

Equation (5) gives the instantaneous error variance as a function of time on an interval between two sample points. The average mean square error on the interval of length τ is found to be

$$\frac{\sigma_{ave}^2}{\sigma_x^2} = \operatorname{ctnh} a\tau - \frac{1}{a\tau} \quad (8)$$

In the special case of uniform sampling, the process of interpolating at each point is equivalent to the optimum reconstruction of $x(t)$ from its samples. This is compared with optimum reconstruction with no time lag for which

$$\frac{\sigma_{ave}^2}{\sigma_x^2} = \frac{2a\tau + e^{-2a\tau} - 1}{2a\tau} \quad (9)$$

For small values of $a\tau$, σ_{ave}^2 is three times as great for Eq. (9) than for Eq. (8).

The problem of interpolating between samples of a Gaussian Markov process has been solved by noting that the samples at both ends of the interval in question are the only data which need be considered. In principle results could also be found for nongaussian Markov process and for vector Gaussian Markov processes. However, in the former case the conditional distributions would make the problem extremely difficult and in the latter case the calculation is conditioned on vector samples at each endpoint, and the results would be difficult to comprehend.

REFERENCES

1. M. G. Kendall and A. Stuart, The Advanced Theory of Statistics, 2nd edition, Vol. 2, pp. 319-321, Hafner Publishing Co., N.Y., 1967.
2. W. Feller, An Introduction to Probability Theory and its Applications, Vol. 2, p. 85, Wiley, N. Y., 1966.
3. G. E. Uhlenbeck and L. S. Ornstein, "On the Theory of Brownian Motion," *Phys. Rev.*, Vol. 36, pp. 823-841, Sept. 1930.

SOME IMPROVED ALGORITHMS FOR COMPUTER SOLUTION OF THE TRAVELING SALESMAN PROBLEM*

K. STEIGLITZ AND P. WEINER

Department of Electrical Engineering, Princeton University
Princeton, New Jersey 08540

ABSTRACT

Two techniques are described for improving Lin's method for obtaining locally optimal solutions to the traveling salesman problem. The first, called dynamic 3-opt, is inductive in nature and produces 3-opt solutions two to three times faster than the basic method described by Lin. The second technique, called linear 2-opt, takes advantage of distance ordering, and requires computation time linear in the number of cities, as opposed to the quadratic time dependence exhibited by the basic method. This approach can be extended to 3-opt and increases the size of problem which can practicably be solved by computers.

INTRODUCTION AND SUMMARY OF PREVIOUS WORK

The traveling salesman problem has received much attention. Briefly, the problem can be stated as follows: Given an $n \times n$ distance matrix (d_{ij}) , determine a permutation of the integers from 1 to n , $(i_1, \dots, i_n)$ which minimizes the sum $d_{i_1 i_2} + d_{i_2 i_3} + \dots + d_{i_n i_1}$. We shall consider only the cases when (d_{ij}) is a symmetric matrix.

Previous work may be divided into two general categories: exact and heuristic.

Exact techniques

Although complete enumeration requires the evaluation of $(n-1)!/2$ tours, Held and Karp [1] and Bellman [2] have shown that the application of dynamic programming reduces the time required for solution from a factorial to an exponential function of n . Little, et.al. [3] and Agin [4] applied a branch and bound method which reduces the time for solution still further, but which still requires exponential time. Other exact techniques include linear programming [5,6], and the method of Croes [7]. The time required by all these algorithms grows so fast with the number of cities that they become impractical for moderate size problems (25-50 cities). For this reason a number of approximate techniques have been developed. Many of the papers referenced above concentrated on approximate applications of the exact techniques. On the other hand there are approximate methods which are not derived from exact techniques, and these will be discussed next.

Heuristic techniques

Karg and Thompson [8] combine two heuristic methods: The first is a fast way to obtain good but not necessarily optimum tours, which works well for convex problems. The second is a way of decomposing the original problem into convex pieces. The

*The research reported here was supported in part by the Bell Telephone Laboratories and in part by the U.S. Army Research Office - Durham under Contract DA-31-124-ARO-D-292.

work of Roberts and Flores [9] is notable in its application of statistical analysis to reduce the size of the problem to one small enough to be completed exhaustively.

In 1965 Reiter and Sherman [10] and Shen Lin [11] suggested the use of fast local optimization techniques applied repeatedly to random initial tours. It should be mentioned that the attainment of the absolute minimum length tour is not the only objective of these methods, which generate many distinct solutions near the optimum. This approach has provided by far the most dramatic reduction in the computation time required to obtain good solutions to large problems. In particular, the computation times documented by Lin [11] represent a great improvement over previously published results, and his approach provides a starting point for our study.

LIN'S METHOD - 3-OPTING

Lin's basic method is to apply a fast but powerful transformation repeatedly to a random tour until no further improvements are possible. The resulting stationary solutions are defined to be locally optimal with respect to the particular transformation. The power of the technique is derived in part from the careful selection of the particular transformation. Lin has also demonstrated that it is possible to extract common features from a collection of distinct locally optimum tours, and that this leads to a reduction technique which considerably reduces computation time.

The particularly successful transformation used by Lin is called 3-opting and is a generalization of the procedure introduced by Croes [7], which can be thought of as 2-opting. In general a tour is called λ -opt [11] if it is not possible to obtain a tour with smaller cost by replacing any λ of its links by any other set of λ links. The transformation involves sequentially examining all ways of interchanging λ links of a given tour, and selecting the first one which produces an improvement.

The overall strategy proposed by Lin consists of starting with random initial tours and obtaining a number of distinct 3-opt tours. By examining common links of these tours, a reduction procedure is applied which commits certain links and effectively reduces the size of the problem for the next round of 3-opting. The reduced problem is attacked in the same way as the original. Although the technique of reduction introduces the danger of improperly committing suboptimum features to the tour at an early stage, it can significantly reduce the time spent on a particular problem.

DYNAMIC 3-OPTING

We shall now describe an alternate method for obtaining 3-opt tours, which we shall call dynamic 3-opting. Experimental results indicate that this method produces 3-opt tours two to three times faster than the basic method proposed by Lin. The method has the property that the 3-opt tours obtained are, on the average, as good as those produced by Lin's 3-opt procedure.

Dynamic 3-opting proceeds as follows, given a starting tour T_0 . A starting city and an orientation are selected, so that T_0 can be characterized by the ordered list of indices $i_1, i_2, \dots, i_n$. The basic idea is to obtain a 3-opt tour for the subproblem consisting of the cities $i_1, \dots, i_{k+1}$ from a 3-opt tour for the subproblem consisting of the cities $i_1, \dots, i_k$. This is done inductively as follows: There is only one tour for the subproblem consisting of the cities i_1, i_2, i_3 , and this must be 3-opt. Now suppose a 3-opt tour has been obtained for the k -city subproblem. An initial tour for the $(k+1)$ -city subproblem is obtained by inserting city i_{k+1} between two consecutive cities of the previously obtained 3-opt tour, and this is done by examining the incremental costs of the k possible tours and choosing the one which results in the minimum cost for the $(k+1)$ -city subproblem. Although the tour obtained in this way is not necessarily 3-opt, experience indicates that it often is. In any event, we need only check those triples of links which include at least one of the two links connected to i_{k+1} . If and when an improvement is obtained by replacing a triple of links with another triple of links, it becomes necessary to check triples which include the newly added links. Thus a list of links is generated which has the property that we need only check triples of links which include members of this list. Proceeding in this way a 3-opt tour is obtained for the $(k+1)$ -city subproblem.

Table 1 shows a comparison of the running times of Lin's 3-opt procedure and dynamic 3-opt on typical problems of size 10 to 57 cities. Both algorithms were coded in Fortran IV with equal sophistication, and run on the CDC 6600 computer. The times shown are the result of averaging 100 3-opt tours from random starts. Also shown are the average percent errors and the number of times correct out of one-hundred trials.

Problem (# of cities)	Dynamic 3-opt			Lin's 3-opt		
	average time in seconds	average % error	number correct per 100	average time in seconds	average % error	number correct per 100
10 ([12])	.038	.6	34	.057	.7	26
20 ([7])	.327	1.9	36	.517	1.5	46
25 ([1])	.378	.7	49	1.17	.75	56
29 ([13])	.683	.9	11	1.72	.9	23
33 ([8])	.956	.75	43	2.63	.6	51
42 ([5])	1.93	.9	31	6.93	1.1	35
48 ([1])	3.06	1.1	8	10.10	1.6	7
48 ([1], [14])	2.96	8.9	5	5.84	11.4	0
57 ([8])	5.31	1.5	0	18.23	1.9	1

Table 1. A comparison of the running times of Dynamic 3-opt and Lin's 3-opt.

During the course of computer experimentation, several heuristic techniques were developed which seemed to work very well on some problems and not so well on others. For example,

an attempt was made to provide initial tours which were effective in combination with dynamic 3-opt. One such method was the nearest neighbor algorithm. In this method of generating an initial tour, a starting city is selected, the city nearest to that is chosen as the next city in the tour, and so on, until a complete tour is generated. Because the beginning of a nearest neighbor tour is more likely to be in the optimum tour than the end, the nearest neighbor tour was fed into the dynamic 3-opting procedure in an order reverse to the one in which it was generated. In this way it was hoped that the poor links would be corrected at an early stage in the optimization. One problem that was especially amenable to this combination of nearest neighbor and dynamic 3-opt was the 48 city Knight's tour, for which the optimum was found 11 times when the nearest neighbor algorithm was started from each of the possible 48 cities. On the other hand, the procedure never produced the correct answer for the 57 city problem although it did produce very good tours. On the 105 city problem [11] the nearest-neighbor-reversed-dynamic 3-opt procedure produced a tour of length 23076, which is lower than any reported previously.

Dynamic 3-opt has the interesting feature that it may be iterated; that is, the tour resulting from one application may be used as an initial permutation for another application, by selecting a starting city and direction. The following strategy was devised to exploit this feature: Once a dynamic 3-opt tour was obtained, it was fed into the dynamic 3-opt procedure with different starting points and directions until a tour with lower cost was found. By repeating this it was found that one could hill-climb to the global optimum. In point of fact, the conjectured global optimum was obtained in this manner for every problem tried, up to and including the 57 city problem. The method has the disadvantage that $2n$ local optima must be checked out at the final stage before it can be determined that no further improvement is possible. Experimental evidence indicates, moreover, that the expected time before obtaining the conjectured global optimum is not significantly less than the time taken by dynamic 3-opting from random starts. This fact is somewhat mitigated by the feature of having a definite termination point.

Another variation of some interest involves the concept of link denial. First a 3-opt tour is obtained. Then a link in the tour is temporarily assigned a high cost, effectively excluding it from future tours. The dynamic 3-opt procedure is then entered at its last stage, since most triplets of the present tour have already been tested. This quickly yields another tour which may or may not be 3-opt relative to the original matrix. This can be checked out by restoring the cost of the denied link to its original value, and by then applying the appropriate tests. The above method provides a very fast procedure for generating a large number of distinct solutions close to the optimum. In fact for the 57 city problem a solution with cost 12978 was found, which is neither globally optimum nor 3-opt, but is the third best solution known. Thus, this procedure may prove of value in engineering applications where non-analytical constraints may be present.

LINEAR 2-OPTING

Dynamic 3-opt, while two to three times faster than Lin's 3-opt, still requires computation time which is asymptotically cubic in the number of cities. If dramatic reduction in the time required to find a local optimum is to be made, means must be found for reducing the asymptotic dependence of time on the number of cities. Such a method will now be described for finding 2-opt tours. By using the concept of local improvement and by ordering the distance matrix, there results a method for 2-opting that takes time asymptotically proportional to n .

Consider a tour with node v_1 connected to branches with lengths a and b . A local improvement is said to exist at v_1 if there exists a link not in the tour terminating at v_1 with length c and if $c < a$ or $c < b$. It can be shown easily that if a 2-opt improvement exists then a local improvement exists. Furthermore, the local improvement determines a favorable 2-change. If now for each node v_1 the other nodes are listed in order of increasing distance from v_1 , we need only search down the list for a local improvement until we have passed the two nodes adjacent to v_1 on the tour. Thus, if the nodes adjacent to v_1 are each marked on the list by a star, we need only search down to the second star. In a problem with a Euclidean metric, it is plausible that the average depth of the second star for a good tour is independent of the number of cities, since it depends only on the local situation at a given city. Indeed, it has been verified experimentally for the class of problems considered here that the average second star depth is 4 to 5. This implies then that a complete checkout for a 2-opt tour requires a time linear in the number of cities, since the search for a local improvement at each city is roughly constant.

The algorithm suggested above requires a starting tour with small average second star depth, for otherwise the initial search for local improvements would still take quadratic time. The following randomized start was used: The cities are ordered randomly; each city, proceeding in the random ordering, is then connected to its nearest neighbor, with the provisions that no city is connected to more than two others, and that no closed cycle is formed until the last step.

The linear 2-opt algorithm with randomized nearest neighbor start was implemented and compared with Lin's 2-opt algorithm with random start, with the results shown in Table 2. Graph 1 shows a plot of computation time vs. problem size, and verifies the linear dependence of the new 2-opt procedure.

Problem # of cities)	Linear 2-opt		Lin's 2-opt	
	average time in seconds	average % error	average time in seconds	average % error
0	.015	1.9	.013	1.6
0	.023	11.1	.038	11.8
5	.028	2.8	.060	3.6
9 ([13])	.035	2.3	.085	3.0
3	.028	2.6	.12	4.6
2	.038	4.5	.19	7.7
8 ([1])	.040	4.3	.25	6.8
8 ([1], [14])	.046	14.9	.12	30.1
7	.052	5.3	.39	6.2

Table 2. A comparison of Linear 2-opt and Lin's 2-opt.

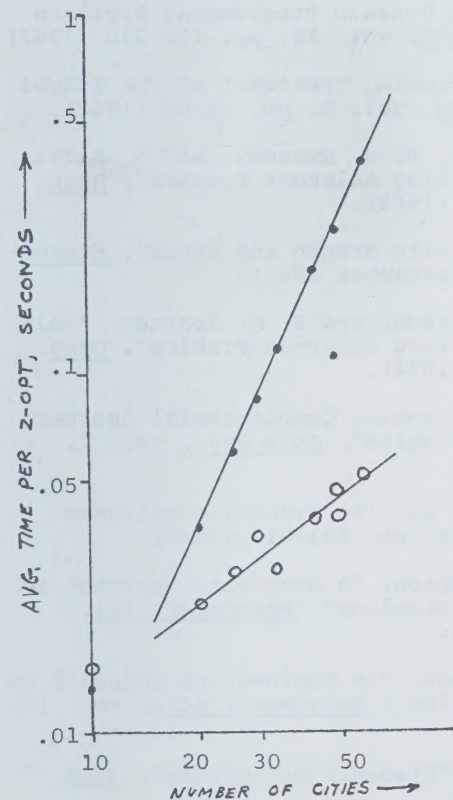


Fig. 1. A comparison of running times for Linear and Lin's 2-opt.

The approach described above has been extended to obtain a local optimization procedure almost as powerful as 3-opt which still takes only linear computation time. Experimental results will be reported in a future paper.

CONCLUSION

We have described two techniques for improving Lin's method for finding locally optimal solutions to the traveling salesman problem. The first, called dynamic 3-opt, is inductive in nature and at each stage adds one city to a 3-opt subproblem and checks only those triples of links which it is necessary to check. The second technique, called linear 2-opt, takes advantage of distance ordering in the search for 2-changes and requires computation time linear in the number of cities, as opposed to the quadratic dependence exhibited by the basic 2-opt algorithm. This approach can be extended to 3-opt and increases considerably the size of problem which can practicably be solved by existing computers.

REFERENCES

1. Held, M. and R. M. Karp, "A Dynamic Programming Approach to Sequencing Problems", SIAM, vol. 10, pp. 196-210 (1962).
2. Bellman, R., "Dynamic Programming Treatment of the Traveling Salesman Problem", J. ACM, vol. 9, pp. 61-63 (1962).
3. Little, J.D.C., K. G. Murty, D. W. Sweeney, and C. Karel, "An Algorithm for the Traveling Salesman Problem", Opns. Res., vol. 11, pp. 979-989 (1963).
4. Agin, N., "Optimum Seeking with Branch and Bound", Management Sci., vol. 13, no. 4 (December 1966).
5. Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson, "Solution of a Large Scale Traveling Salesman Problem", Opns. Res., vol. 2, pp. 393-410 (1954).
6. _____, "On a Linear Programming Combinatorial Approach to the Traveling Salesman Problem", Opns. Res., vol. 7, pp. 58-66 (1959).
7. Croes, G. A., "A Method for Solving Traveling Salesman Problems", Opns. Res., vol. 6, pp. 791-812 (1958).
8. Karg, L. L., and G. L. Thompson, "A Heuristic Approach to Solving Traveling Salesman Problems", Management Sci., vol. 10, pp. 225-248 (1964).
9. Roberts, S. M., and B. Flores, "An Engineering Approach to the Traveling Salesman Problem", Management Sci., vol. 13, pp. 269-288 (1966).
10. Reiter, S. and G. Sherman, "Discrete Optimizing", SIAM, vol. 13, pp. 864-889 (1965).

11. Lin, S., "Computer Solution of the Traveling Salesman Problem", BSTJ, vol. 44, pp. 2245-2269 (1965).
12. Barachet, L. L., "Graphic Solution of the Traveling Salesman Problem", Opns. Res., vol. 5, pp. 841-845 (1957).
13. Lin, S., private communication.
14. Knight's Tour.

TOPOLOGICAL FORMULATION OF NETWORK FUNCTIONS
WITHOUT GENERATION OF K-TREES

W.R. DUNN, Jr.⁺ and S. P. CHAN⁺⁺
University of Santa Clara, Santa Clara, California

ABSTRACT

A recognized limitation to the application of topological formulas to linear network analysis is the need to find all k-trees at the network graph which number may be very large.

Using early ideas of Percival [1] and Mayeda [2] the paper presents a method for deriving k-tree terms (i.e. partial sums of k-tree admittance products) of an arbitrary graph. It is shown that the number of terms is significantly less than the number of trees.

The use of k-tree terms in the topological analysis of networks results in both a more compact notation when symbolic results are sought and permits the handling of larger networks than can be practically accommodated solely by generation of individual k-trees.

INTRODUCTION

Traditionally, the approach to using topological formulas in the analysis of linear networks has been to generate k-trees of the network graph. It is well known that this approach places a limit on the size of a network which may be handled by such means due to the rapid growth in the number of k-trees as the network graph becomes large.

In a classical paper Percival [1] has described means for "decomposing" network graphs into simpler forms without the need to generate individual k-trees of the graph. While Percival describes the decomposition of specific types of network graphs, Mayeda in a later correspondence [2] has discussed a more general approach. Specifically, Mayeda discusses the "partitioning" of the graph into subgraphs, finding topological quantities for each, and recombining these to produce the tree admittance products of the graph.

The following paragraphs discuss an approach based on Mayeda's work which permits the topological formulation of passive network functions without generating individual k-trees of a graph.

DEFINITIONS AND NOTATION

The definitions used in this paper but not cited below can be found in Seshu and Reed [3] and Chan [4].

Consider a connected graph G of v vertices. Next, remove a set of edges* of G such that a subgraph G' is obtained which is in k connected pieces.

⁺Also with Lockheed Missiles and Space Company, Sunnyvale, California

⁺⁺ Also supported in part by the National Aeronautics and Space Admin.
Grant No. NGR-05-017-017

* But not isolated vertices where they arise in the process of removing the edges.

Definition 1. Connected Vertex Set. The vertices in each connected piece of G' are called a connected vertex set. It should be noted that a connected vertex set may comprise an isolated vertex.

Definition 2. Vertex Partition Set. The set of all connected vertex sets of G' is defined as a vertex partition set. It is convenient to consider the labelling of the v vertices of G (and hence G') symbolically as $v_1, v_2, v_3, \dots, v_n$. The vertex partition set of G' may be denoted using these symbols as shown through the following example.

Example #1: Consider the connected, vertex labelled, graph G of Fig. 1a. Remove edges of G_1 to obtain G_1' as shown in Fig. 1b. Looking at G_1' in Fig. 1b, the vertex sets $\{v_1, v_2, v_3\}, \{v_4, v_5\}, \{v_6, v_7\}, \{v_8\}$ are connected vertex sets. The set of all these in turn constitutes a vertex partition set which may be denoted by $\{v_1, v_2, v_3; v_4, v_5; v_6, v_7; v_8\}$ with semicolons separating the symbolic groups corresponding to the connected vertex sets of the graph. Returning to the case of subgraph G' of G , it is clearly seen that G' which is in k -connected pieces includes one or more k -trees of G . That is, G' is either a k -tree or by removal of one or more of its edges, a k -tree of G may be formed.

Definition 3. k -Tree With Respect to a Vertex Partition Set. Consider G' having k connected vertex sets $\{K_1\}, \{K_2\}, \dots, \{K_k\}$ comprising a vertex partition set $\{K_1; K_2; \dots; K_k\}$. A k -tree of G formed by removal of a set of edges from G' may be associated with the vertex partition set in the notation $T_{G'}^n \{K_1; K_2; \dots; K_k\}$, defined as a k -tree with respect to vertex partition set $\{K_1; K_2; \dots; K_k\}$. Since it may be possible to form more than one k -tree by removal of edges in G' , the superscript " n " in the notation corresponds to an integer which permits several such k -trees to be distinguished. (Subscript G' means that the k -tree of G is formed from subgraph G'). Since the k -tree comprises a subset of the edges of G , it is useful to next label the edges of G to correspond to the branch admittances of the corresponding network of G . Symbol y_{ij} accordingly will denote the admittance value connecting vertices v_i and v_j ; that is, y_{ij} will represent combined parallel edges where they exist in the network.

Definition 4. k -tree term With Respect to Vertex Partition Set. Consider again G' comprising vertex partition set $\{K_1; K_2; \dots; K_k\}$ from which k -trees $T_{G'}^1 \{K_1; K_2; \dots; K_k\}, T_{G'}^2 \{K_1; K_2; \dots; K_k\}, \dots, T_{G'}^n \{K_1; K_2; \dots; K_k\}$ can be formed. By forming the product of admittances of each such k -tree* and summing these products for all n k -trees, a k -tree term with respect to vertex partition set $\{K_1; K_2; \dots; K_k\}$ is formed. It will be denoted by $V_{G'} \{K_1; K_2; \dots; K_k\}$ where, $V_{G'} \{K_1; K_2; \dots; K_k\} = \sum_{i=1}^n$ (admittance product of $T_{G'}^i \{K_1; K_2; \dots; K_k\}$), the subscript G' denoting subgraph G' of G .

In the following paragraphs, Definitions 3 and 4 will be abbreviated to k -tree and k -tree term (or simply term) respectively where the vertex partition set is clearly implied. All the foregoing definitions are illustrated in the following example.

Example #2: Consider the connected, edge and vertex labelled Graph G_2 in Fig. 2a. By removal of edges y_{24} and y_{34} , subgraph G_2' as shown in Fig. 2b is formed. The connected vertex sets in G_2' are $\{v_1, v_2, v_3\}$ and $\{v_4\}$. This set in turn comprises vertex partition set $\{v_1, v_2, v_3; v_4\}$. The 2-trees (i.e. $k = 2$) may be denoted.

* Where G' comprises only isolated vertices the v -tree admittance product is defined as unity.

$T_{G_2}^1 \{v_1 v_2 v_3; v_4\}$ edges y_{12} & y_{13} ; $T_{G_2}^2 \{v_1 v_2 v_3; v_4\}$ edges y_{12} & y_{23}
 $T_{G_2}^3 \{v_1 v_2 v_3; v_4\}$ edges y_{13} & y_{23}

Finally, the 2-tree term may be written:

$$V_{G_2} \{v_1 v_2 v_3; v_4\} = y_{12} y_{13} + y_{12} y_{23} + y_{13} y_{23}$$

It is important to note that these above definitions use a similar, but not identical notation to that of Mayeda [1]. In particular, the notation in this paper corresponding to a vertex partition set includes all the vertices of the graph G whereas the notation used in Mayeda's work refers in general to a subset of the vertices of the graph. The reason for such variance is to remove the restriction of specifying such a subset in the subsequent discussion.

THEORETICAL DEVELOPMENT

Mayeda [1] has discussed the "partitioning" of a connected graph G of v vertices into complementary subgraphs G_1 and G_2 such that edges of G_1 have $v' \leq v$ vertices in common with the edges of G_2 . The following lemma provides a generalization of Mayeda's criteria for realizing a tree of the graph so partitioned.

Lemma: Consider a graph G of v vertices having complementary connected subgraphs G_1 and G_2 such that edges of G_1 have $v' \leq v$ vertices in common with the edges of G_2 . A k -tree t_k of G may be formed from an i -tree ($k \leq i \leq v$) t_i of G_1 only if it is combined with a $(k + v' - i)$ - tree $t_{k+v'-i}$ of G_2 .

Proof: Note first that the tree t_k is in k connected pieces. Also, the vertices in each connected piece in t_i may appear in no other connected piece in t_i (otherwise it would be connected to the other piece); similarly, the vertices in each connected piece in $t_{k+v'-i}$ may appear in no other connected piece in $t_{k+v'-i}$. Now t_i and $t_{k+v'-i}$ together are in $(i) + (k + v' - i) = (k + v')$ pieces. If the pieces have no vertices in common (i.e., $v' = 0$), the number of pieces that result by combining them is the original number of pieces, $k + 0 = k$. If t_i and $t_{k+v'-i}$ have one vertex in common (i.e., $v' = 1$) then upon combining all the pieces, a resulting graph is obtained having one less connected piece or exactly $(k + 1 - 1) = k$ pieces. Next, observe the case where t_i and $t_{k+v'-i}$ have more than one vertex in common. Now no piece in G_1 may have more than one vertex in common with any piece in G_2 otherwise by recombining the pieces a circuit would result (and hence not a tree). Consequently, if the pieces are recombined by coalescing the common vertices one at a time, the number of original pieces is reduced exactly by v' and the remaining number of pieces is $[(k + v') - (v')] = k$.

With these observations, the lemma is proved by contradiction by assuming that for $p \neq k$, t_i and $t_{p+v'-i}$ produce a k -tree.

Now, by letting both G_1 and G_2 contain all the vertices of G (i.e., G_1 and G_2 may contain isolated vertices) it is clearly seen that $v' = v$. Utilizing the definition of a k -tree as cited in Def. 3, but momentarily disassociating it with vertex partition set, the lemma may be restated as follows:

*Lower case t is used in this lemma to denote k -trees of graph G , and subgraphs G_1 and G_2 . Upper case T as stated in Def. 3, on the other hand, denotes k -tree of graph G formed from subgraph G' of G .

Corollary: Consider a connected graph G of v vertices having complementary subgraphs G_1 and G_2 , both of which contain all the vertices of G . A k -tree T_G of G may be formed from an i -tree T_{G_1} of G only if it is combined with a $(k+v-i)$ - tree T_{G_2} of G .

Although the Lemma and Corollary state only necessary conditions, sufficient conditions are implied in the proof of the Lemma wherein it is noted that no connected piece of G_1 may have more than one vertex in common with any other connected piece in G_2 . This observation, along with the lemma and corollary lead to the following theorem.

Theorem: Consider a connected graph G of v vertices having complementary subgraphs G_1 and G_2 , both of which contain all the vertices of G . Let $T_{G_1}\{K_i\}$ be an i -tree of G with respect to vertex partition set K . A k -tree of G may be formed from $T_{G_1}\{K_i\}$ if, and only if, it is combined with a $(k+v-i)$ tree of G , $T_{G_2}\{K_{k+v-i}\}$, with respect to vertex partition set K_{k+v-i} , provided that no two connected vertex sets in K_i and K_{k+v-i} have more than one vertex in common.

The theorem is illustrated in the following example:

EXAMPLE #3: Consider the labelled graph G shown in Fig. 3(a) with its decomposition into complementary subgraphs G_1 and G_2 shown respectively in Figs. 3(b) and 3(c).

Now, to form a tree (i.e. $k = 1$) of G from a 3-tree (i.e. $i = 3$) of G formed from G_1 it is necessary to combine the 3-tree with a $(k+v-i)=2$ -tree of G formed from G_2 . Referring to Fig. 3(b), 3-tree $T_{G_1}\{v_1 v_4; v_2; v_3\}$ in G_1 and 2-tree $T_{G_2}\{v_1; v_2 v_3 v_4\}$ in G_2 do form a tree of G since no two connected vertex sets have more than one vertex in common. On the other hand, $T_{G_1}^2\{v_1; v_2 v_4; v_3\}$ and $T_{G_2}^1\{v_1; v_2 v_3 v_4\}$ do not form a tree since connected vertex set $\{v_2 v_4\}$ in $T_{G_1}^2\{v_1; v_2 v_4; v_3\}$ has more than one vertex in common with $\{v_2 v_3 v_4\}$ in $T_{G_2}^1\{v_1; v_2 v_3 v_4\}$. On the other hand, a 2-tree of G may be formed by combining $T_{G_1}^1\{v_1; v_2 v_4; v_3\}$ with $T_{G_2}^2\{v_1; v_4; v_2 v_3\}$. In the foregoing example, a 2-tree $T_{G_1}^1\{v_1 v_2 v_4; v_3\}$ and 3-tree $T_{G_2}^1\{v_1; v_4; v_2 v_3\}$ would combine to yield a tree of G . Clearly, a 2-tree admittance product $T_{G_1}^1\{v_1 v_2 v_4; v_3\}(y)^*$ multiplying $T_{G_1}^1\{v_1; v_4; v_2 v_3\}(y)$ yields a tree admittance product of G . Correspondingly, the sum of all 2-tree admittance products corresponding to $\{v_1 v_2 v_4; v_3\}$ in G_1 multiplying the sum of all 3-tree admittance products corresponding to $\{v_1; v_4; v_2 v_3\}$ in G_2 yield a (partial) sum of tree admittance products of G . Referring to Definition 4, this observation may be more concisely described by stating that a (partial) sum of tree admittance products of G is obtained by forming the product of the 2-tree term $V_{G_1}\{v_1 v_2 v_4; v_3\}$ and the 3-tree term $V_{G_2}\{v_1; v_4; v_2 v_3\}$.

Example #4: Referring again to Fig. 3.

$$\begin{aligned}
 V_{G_1}\{v_1 v_2 v_4; v_3\} &= \sum_{\substack{\text{(all 2-tree admittance products} \\ \text{w.r.t. } \{v_1 v_2 v_4; v_3\})}} T_{G_1}^m\{v_1 v_2 v_4; v_3\}(y) \\
 &= T_{G_1}^1\{v_1 v_2 v_4; v_3\}(y) + T_{G_1}^2\{v_1 v_2 v_4; v_3\}(y) + T_{G_1}^3\{v_1 v_2 v_4; v_3\}(y) \\
 &= y_{12}y_{14} + y_{14}y_{24} + y_{12}y_{24} \\
 V_{G_2}\{v_1; v_4; v_2 v_3\} &= T_{G_2}^1\{v_1; v_4; v_2 v_3\}(y) = y_{23}
 \end{aligned}$$

* i.e. the (y) means admittance product.

Multiplying,

$$V_{G_1}\{v_1v_2v_4;v_3\} \cdot V_{G_2}\{v_1;v_4;v_2v_3\} = (y_{12}y_{14} + y_{14}y_{24} + y_{12}y_{24}) \cdot y_{23} \quad (1)$$

It is seen that (1) represents a (partial) sum of tree admittance products of G. It is important at this point to note that (1) is not expanded to read:

$$V_{G_1}\{v_1v_2v_4;v_3\} \cdot V_{G_2}\{v_1;v_4;v_2v_3\} = y_{12}y_{14}y_{23} + y_{14}y_{24}y_{23} + y_{12}y_{24}y_{23} \quad (2)$$

The reasoning for this preference is twofold. First, as pointed out by Percival [1], the notation given by (1) is more concise than that of (2). Second, where the y_{ij} are numbers (i.e. branch conductances) the numerical value of term products is obtained with fewer arithmetic operations by using (1).

Although such economies do not appear justified by the added notational complexity for this simple example, they become significant as the size of the graph becomes large. This is borne out in the following paragraphs.

GENERATION OF k-TREE TERMS OF AN ARBITRARY CONNECTED GRAPH

Consider a passive network with corresponding labelled graph G of v vertices. On the basis of the foregoing definitions and theorem, it is seen that the k-tree terms of G may be derived through the following procedure:

1. Identify edge y_{12} connecting vertices v_1 and v_2 . Let this edge and the remaining (isolated) vertices of G constitute subgraph G_2^1 of G. Note that G_2^1 is a (v-1)-tree, $T_{G_2^1}\{v_1v_2;v_3...;v_v\}$ of G. The corresponding (v-1)-tree term is $V_{G_2^1}\{v_1v_2;v_3...;v_v\} = y_{12}$

2. Next identify another vertex v_j which has one or two edges incident with respectively one or both of vertices v_1 and v_2 . (such a vertex may always be found since G is connected). Let these edges and the vertices of G constitute subgraph G_1^j of G. Next form from all possible combinations of edges of G_1^j all possible k-tree terms of G. In this instance, it is seen that three possible situations exist:

Case (a): Vertex v_j is connected only by edge y_{1j} to vertex v_1 . The (v-1)-tree term is: $V_{G_1^j}\{v_1v_j;...;v_v\} = y_{1j} \quad (v \neq j)$

Case (b): Vertex v_j is connected only by edge y_{2j} to vertex v_2 . The (v-1)-tree term is: $V_{G_1^j}\{v_2v_j;...;v_v\} = y_{2j} \quad (v \neq j)$

Case (c): Vertex v_j is connected by edges y_{1j} and y_{2j} to vertices v_1 and v_2 respectively. There are two (v-1)-tree terms, $V_{G_1^j}\{v_1v_j;...;v_v\} = y_{1j}$; $V_{G_1^j}\{v_2v_j;...;v_v\} = y_{2j}$, and one (v-2)-tree term, $V_{G_1^j}\{v_1v_2v_j;...;v_v\} = y_{1j}y_{2j}$

3. Perform the union of subgraphs G_2^1 and G_1^j and designate the resultant subgraph as G_2^2 . Where the theorem permits, combine terms of G_1^j with those of G_2^1 to form all possible k-tree terms ($v-2 \leq k \leq 1$) of G. Where a term of G_1^j and terms resulting from the combination of terms of G_1^j and G_2^1 have the same vertex partition set, combine (i.e. add) these terms to form a single term of G_2^2 with the given vertex partition set. Retain these and other previously generated terms but letting all terms correspond to G_2^2 .

4. Next, identify vertex v_k which is connected to one or more of vertices v_1, v_2 , and v_4 and those edges connecting v_k to one or more of these latter vertices. Let these edges and all vertices of G constitute subgraph G_2^2 of G . Using only G_2^1 , find all possible k -tree terms of G . Perform the union of subgraphs G_2^1 and G_2^2 and denote the resultant subgraph by G_2^3 . Where the theorem permits, combine terms of G_2^1 with those of G_2^2 to form all possible distinct k -tree terms ($v-2 < k < l$) of G (i.e. terms having distinct vertex partition sets). In every instance that a term of G_2^1 and a term resulting from the combination of terms of G_2^1 and G_2^2 have the same vertex partition set, combine (i.e. add) these terms to form a single term of G_2^3 with the given vertex partition set. Retain these and other terms in G_2^2 as well as those terms retained in Step 3 but letting all terms correspond to G_2^3 .

5. Continue this basic process of "introducing" vertices one at a time but only where each vertex has at least one edge connecting it to previously introduced vertices. Designate the edges connecting this vertex to previously introduced vertices and the vertices of G as subgraph G_2^m . Using only G_2^m , find all possible k -tree terms of G . Perform the union of G_2^m and G_2^{m-1} (formed in the previous step) designating the resultant subgraph as G_2^{m+1} . Where the theorem permits, combine terms of G_2^m with those of G_2^{m-1} to form all possible distinct k -tree terms ($v-2 < k < l$) of G . In every instance that a term of G_2^m and a term, resulting from the combination of terms of G_2^m and G_2^{m-1} have the same vertex partition set, combine (i.e. add) these terms to form a single term of G_2^{m+1} with the given vertex partition set. Retain these and other terms retained in the previous step but letting all terms correspond to G_2^{m+1} . The process terminates when all vertices have been introduced and subgraph $G_2^{v-1} \equiv G$ is realized. The retained terms correspond to the k -trees of G . This procedure is illustrated in the following example:

Example #5: Consider again the labelled graph G shown in Figure 3a.

1. Identify edge y_{12} connecting vertices v_1 and v_2 . Let this edge and the remaining (isolated) vertices of G constitute subgraph G_2^1 as shown in Fig. 3d. Note that G_2^1 is a 3-tree of G with the corresponding 3-tree term

$$V_{G_2^1}\{v_1 v_2; v_3; v_4\} = y_{12}$$

2. Next identify vertex v_4 which has edges y_{14} and y_{24} incident with vertices v_1 and v_2 respectively. Let these edges and vertices of G constitute subgraph G_2^1 as shown in Figure 3e. Using all possible combinations of edges of G_2^1 form the following k -tree terms of G : $V_{G_2^1}\{v_1 v_4; v_2; v_3\} = y_{14}$, $V_{G_2^1}\{v_2 v_4; v_1; v_3\} = y_{24}$, $V_{G_2^1}\{v_1 v_2 v_4; v_3\} = y_{14} y_{24}$. Note that this example corresponds to Case (c) in Step 2 of the procedure.

3. Perform the union of G_2^1 and G_2^1 designating the resultant subgraph G_2^2 as shown in Fig. 3f. Using the theorem, combine terms of G_2^1 with those of G_2^1 to realize the following terms of G :

$$[V_{G_2^1}\{v_1 v_4; v_2; v_3\} + V_{G_2^1}\{v_2 v_4; v_1; v_3\}] \cdot V_{G_2^1}\{v_1 v_2; v_3 v_4\} = (y_{14} + y_{24}) \cdot y_{12}$$

Now this term results in a term with vertex partition set $\{v_1 v_2 v_4; v_3\}$. A term of G_2^1 was generated above having the same vertex partition set, $V_{G_2^1}\{v_1 v_2 v_4; v_3\} = y_{14} y_{24}$. Consequently, the foregoing two terms may be combined to form $V_{G_2^2}\{v_1 v_2 v_4; v_3\} = y_{14} y_{24} + (y_{14} y_{24}) y_{12}$

This term will be retained along with, $V_{G_2}^2\{v_1v_2;v_3;v_4\} = V_{G_1}^1\{v_1v_2;v_3;v_4\} = y_{12}$
 $V_{G_2}^2\{v_1v_4;v_2;v_3\} = V_{G_1}^1\{v_1v_4;v_2;v_3\} = y_{14}$, $V_{G_2}^2\{v_2v_4;v_1;v_3\} = V_{G_1}^2\{v_2v_4;v_1;v_3\} = y_{24}$

4. Next identify vertex v_3 which is connected to vertices v_2 and v_4 by edges y_{23} and y_{34} respectively. Let these edges and vertices of G_2 constitute subgraph G_1^2 as shown in Fig. 3f. Using G_1^2 , find the following k-tree terms of G , $V_{G_1}^2\{v_2v_3;v_1;v_4\} = y_{23}$, $V_{G_1}^2\{v_3v_4;v_1;v_2\} = y_{34}$, $V_{G_1}^2\{v_2v_3v_4;v_1\} = y_{23}y_{34}$

Perform the union of G_1^2 and G_2^2 designating the resultant subgraph G_3^2 . Note at this point that no more vertices are to be introduced hence $G_3^2 = G_2^2$. Using the theorem, combine terms of G_1^2 with terms of G_2^2 to obtain the following k-tree terms of G : $V_{G_1}^2\{v_2v_3;v_1;v_4\} \cdot V_{G_2}^2\{v_1v_2;v_3;v_4\} = y_{23}y_{12}$,
 $V_{G_1}^2\{v_2v_3;v_1;v_4\} \cdot V_{G_2}^2\{v_1v_4;v_2;v_3\} = y_{23}y_{14}$, $[V_{G_1}^2\{v_2v_3;v_1;v_4\} + V_{G_2}^2\{v_3v_4;v_1;v_2\}] \cdot V_{G_2}^2\{v_2v_4;v_1;v_3\} = (y_{23} + y_{34})y_{24}$, $V_{G_1}^2\{v_3v_4;v_1;v_2\} \cdot V_{G_2}^2\{v_1v_4;v_2;v_3\} = y_{34}y_{14}$
 $[V_{G_1}^2\{v_2v_3;v_1;v_4\} + V_{G_2}^2\{v_3v_4;v_1;v_2\}] \cdot V_{G_2}^2\{v_1v_2v_4;v_3\} + V_{G_1}^2\{v_2v_3v_4;v_1\} \cdot [V_{G_2}^2\{v_1v_2;v_3;v_4\} + V_{G_2}^2\{v_1v_4;v_2;v_3\}] = (y_{23} + y_{34}) \cdot [y_{14}y_{24} + (y_{14} + y_{24})y_{12}] + y_{23}y_{34}(y_{12} + y_{14})$.

Now the third term in the above may be combined with G_1^2 term $V_{G_1}^2\{v_2v_3v_4;v_1\} = y_{23}y_{34}$ to yield $V_G\{v_2v_3v_4;v_1\} = (y_{23} + y_{34}) \cdot y_{24} + y_{23}y_{34}$ since both have the same vertex partition set. The last term is the tree term of G and may be rewritten, $V_G\{v_1v_2v_3v_4\} = (y_{23} + y_{34}) \cdot [y_{14}y_{24} + (y_{14} + y_{24})y_{12}] + y_{23}y_{34}(y_{12} + y_{14})$.

From the above results, the following 2-tree terms of G may be written:
 $V_G\{v_1v_2v_3;v_4\} = y_{23}y_{12}$, $V_G\{v_2v_3;v_1v_4\} = y_{23}y_{14}$, $V_G\{v_3v_4;v_1v_2\} = y_{34}y_{12}$,
 $V_G\{v_1v_3v_4;v_2\} = y_{34}y_{14}$, $V_G\{v_2v_3v_4;v_1\} = (y_{23} + y_{34}) \cdot y_{24} + y_{23}y_{34}$,
 $V_G\{v_1v_2v_4;v_3\} = y_{14}y_{24} + (y_{14} + y_{24})y_{12}$. Finally, the 3-tree terms of G may be written, $V_G\{v_1v_2;v_3;v_4\} = y_{12}$, $V_G\{v_2v_4;v_1;v_3\} = y_{24}$, $V_G\{v_1v_4;v_2;v_3\} = y_{14}$,
 $V_G\{v_2v_3;v_1;v_4\} = y_{23}$, $V_G\{v_3v_4;v_1;v_2\} = y_{34}$. It is seen that all k-tree terms of G have been obtained.

ENUMERATION OF TERMS OF THE COMPLETE GRAPH

As noted earlier, the problem of obtaining a very large number of trees of a relatively large graph can be reduced to one of finding a significantly smaller number of terms.

For a complete graph G of v vertices, it is known that the number of trees is given by v^{v-2} [5]

Since every pair of vertices in G is connected by an edge, it is clearly seen that the number of terms in G is exactly the number of all possible vertex partition sets that can be formed from the v vertices less one (since the vertex partition set $\{v_1;v_2;v_{v-1};v_v\}$ is not generated).

Now it is known [6] that the number of ways of grouping n distinct items into m groups with no group empty is given by the Stirling number of second

kind $S(n,m)^*$. Accordingly, the number of terms in G is given by
$$\sum_{m=1}^{v-1} S(v,m)$$

Table 1 shows the number of trees and terms (generated from the foregoing procedure) in terms of the vertices of the complete graph. Although the number of terms grows rapidly with the size (i.e. number of vertices) of the complete graph, it is readily seen that for a given number of vertices the number of terms is indeed far less than the number of trees. Equally important, the terms moreover correspond to all needed k -trees of the graph.

Although Table 1 corresponds to a complete graph, it has been empirically observed [7] that the number of k -trees (vs. nodes) of practical networks increases in roughly the same manner. For this reason, limitations in the topological-formulae approach due to the number of trees would appear substantially ameliorated through derivation of k -tree terms.

CONCLUSIONS

The paper has elaborated the early notions of Mayeda^[2], presented an approach for deriving k -tree terms at an arbitrary connected graph, and finally on an enumerative basis postulated the superiority of the use of k -tree terms over individual k -trees.

In practical engineering application, and in particular when digital computers are employed, the approach proffers two important advantages:

1. Where symbolic transfer function expressions are sought, the k -tree terms provide a more compact notation.
2. The maximum size of a network that can be analyzed by the topological formulae is significantly increased.

Extensions of this work include application to active networks, optimal "partitioning" (i.e., derivation and recombination of terms of subgraphs of the graph as originally proposed by Mayeda), and finally their implementation by digital computer.

* $S(n,m)$ is normally given in tables generated from recursion relationship $S(n+1,k) = S(n,k-1) + kS(n,k)$ where $S(1,1) = S(2,1) = S(2,2) = 1$

REFERENCES

- [1] W. S. Percival, "The Solution of Passive Electrical Networks by Means of Mathematical Trees", Institution of Electrical Engineers, Part 3, Vol. 100, May 1953.
- [2] W. Mayeda, "Reducing Computation Time in the Analysis of Networks by Digital Computer", IRE Transactions on Circuit Theory, Vol. CT-6, March 1959.
- [3] Seshu and Reed, Linear Graphs and Electrical Networks, Addison-Wesley, 1961.
- [4] S. P. Chan, Introductory Topological Analysis of Electrical Networks, Holt, Rinehart & Winston, 1969 (to be published).
- [5] R. Busacker and T. Saaty, Finite Graphs and Networks, McGraw-Hill, 1965.
- [6] J. Riordan, An Introduction to Combinatorial Analysis, Wiley & Sons, 1958.
- [7] D. Dawson, "Computational Aspects of the Topological Approach to Active Linear Network Analysis", Proc. of Hawaii International Conference on System Sciences, 1968.

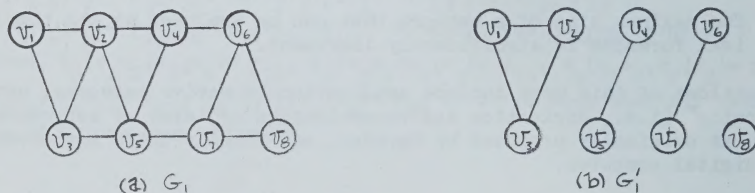


Figure 1

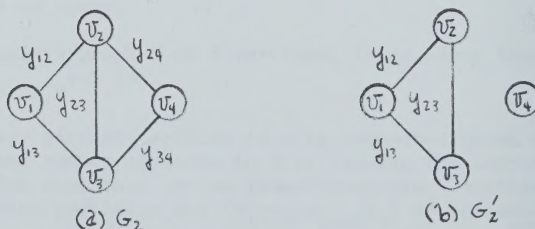


Figure 2

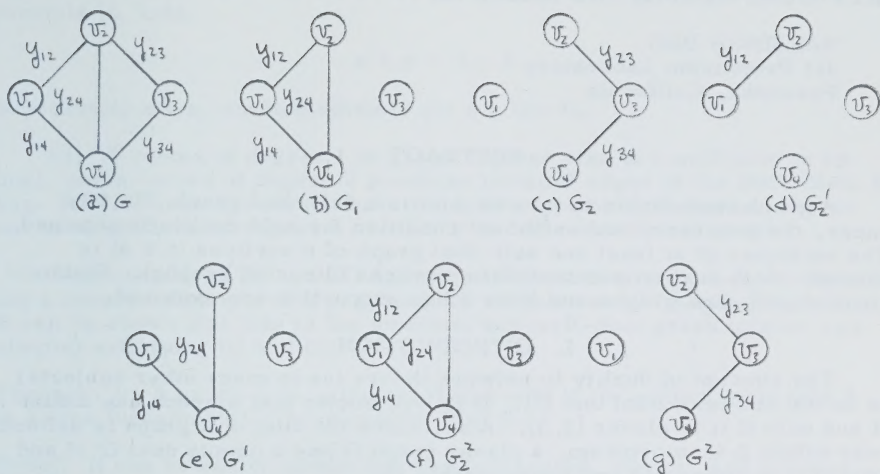


Figure 3

TABLE 1: Terms and Trees vs. Vertices for the Complete Graph

No. of Vertices	No. of Trees(v^{v-2})	No. of 1-, 2-, & 3- tree terms($\sum_{m=1}^3 S(v, m)$)
4	16	14
5	125	41
6	1296	122
7	16807	365
8	262144	1094
9	4.78×10^6	3281
10	10^8	9842
15	1.95×10^{15}	2.39×10^6
20	2.62×10^{23}	5.8×10^8
25	1.42×10^{32}	1.4×10^{11}
30	2.29×10^{41}	3.4×10^{13}

SELF-DUAL GRAPHS AND DIGRAPHS*

NARSINGH DEO
Jet Propulsion Laboratory
Pasadena, California

ABSTRACT

A graph isomorphic to its own dual is a self-dual graph. In this paper, the necessary and sufficient condition for self-duality is obtained. The existence of at least one self-dual graph of n vertices ($n \geq 4$) is proved. Self-duality is extended to digraphs (directed graphs). Enumeration of self-dual graphs and their other properties are explored.

I. INTRODUCTION

The concept of duality in network theory (as in many other subjects) is an old and important one [1]. It is well known that a graph has a dual if and only if it is planar [2, 3]. Also, since the dual of a graph is defined only within 2-isomorphism, a planar graph G has a unique dual G' if and only if G is connected and nonseparable [3].

Furthermore, by graphs here we shall imply only those graphs that have no self-loop (i. e., an edge joining a node to itself) and no parallel edges (i. e., more than one edge joining the same node-pair); i. e., simple graphs. Thus any meaningful discussion of self-duality must be limited to planar, simple, connected, nonseparable, linear graphs.

Definition: A planar, connected, nonseparable, simple graph G is called self-dual if graph G is isomorphic to its dual G' . For example, graphs in Figs. 1, 2, and 3 are self-dual.

Although some work has been done on self-dual configurations (duality being defined between lines and points, rather than between nodes and meshes, as is done for graphs) [4, 5], the properties of self-dual graphs have not been investigated. As we will see shortly, self-duality is a very interesting property, and we confront many challenging unsolved problems connected with it.

II. PROPERTIES OF SELF-DUAL GRAPHS

Theorem 1: The following two conditions are necessary (but not sufficient) for a graph G to be self-dual:

- (i) The number of edges in G is $e = 2(n-1)$, where n is the number of vertices in G .
- (ii) The degree $d(v_i) \geq 3$, for every vertex v_i of G .

Proof: (i) Let G be a self-dual graph. Then the number of regions, r , (windows, meshes, or faces) of G , when G is mapped on a sphere, must be

*This paper presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract No. NAS 7-100, sponsored by the National Aeronautics and Space Administration.

equal to the number of its vertices, n , i.e., $r = n$. But from Euler's formula [6, 7, 8],

$$n + r = e + 2$$

Substituting $r = n$, we immediately get $e = 2(n-1)$.

(ii) A vertex of degree 1 in graph G gives rise to a self-loop in its dual, and a vertex of degree 2 produces parallel edges in the dual [Ref. 8, Fig. 4.7]. Since G does not have parallel edges or self-loops, it must not have vertices of degree 1 or 2, if it is to be a self-dual graph.

That conditions (i) and (ii) are not sufficient can be seen by constructing a counter-example of a graph with 7 vertices and 12 edges (Fig. 4). It can be shown that this is the smallest non-self-dual graph (planar and simple) satisfying (i) and (ii).

Theorem 2: A graph G is self-dual if and only if its incidence matrix A is also a fundamental-circuit matrix B of G .

Proof: It can be easily shown that two (connected, planar, and nonseparable) graphs are dual of each other if and only if the incidence matrix of one is the fundamental circuit matrix of the other (if the meshes are chosen as the circuits) [Ref. 3, p. 85]. It immediately follows that a graph G is self-dual if and only if the incidence matrix of G is also its fundamental circuit matrix.

Theorem 2 can be illustrated by the following example, where A = incidence matrix and B = circuit matrix, of the self-dual graph of Fig. 2.

$$A = \begin{matrix} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

$$B = \begin{matrix} & \begin{matrix} e_2 & e_1 & e_6 & e_7 & e_8 & e_3 & e_4 & e_5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

In writing A , we took all vertices into account and did not leave out the row corresponding to the reference vertex, as is done by most authors when they refer to an incidence matrix. Similarly, in writing the circuit matrix we took into account all meshes including the outside circuit--i.e., the circuit defining the unbounded region. One notices also that the order of edges in B is not the same as that in A .

Corollary: Let B' be the $n \times 2(n-1)$ matrix obtained by permuting the columns of B , so that edges in B' are ordered in the same way as in A . Then we know that $B'A^t = 0 = A B'^t$, because of the orthogonality of circuit and

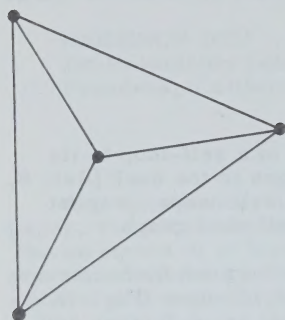


Fig. 1. Self-dual graph of 4 vertices

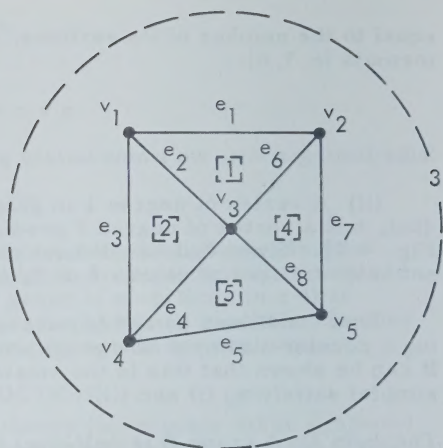


Fig. 2. Self-dual graph of 5 vertices

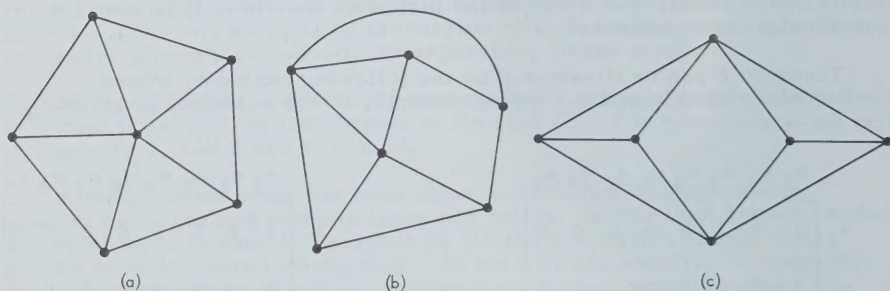


Fig. 3. Self-dual graphs of 6 vertices

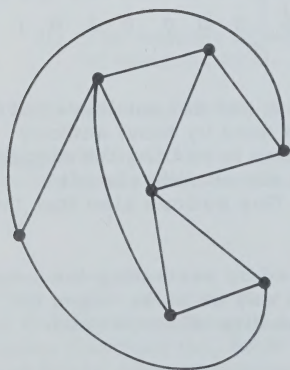


Fig. 4. The smallest non-self-dual planar graph satisfying (i) and (ii) of Theorem 1

incidence row vectors.* In other words, a graph G (simple, planar, connected, etc.) is self-dual if and only if there exists a matrix A' obtained by column permutations of A , such that $A'A^t = 0 \pmod{2}$.

This property is illustrated by referring to Fig. 2 again:

$$B' = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

It is easy to verify that $B'A^t = AB'^t = 0 \pmod{2}$. The permutation of rows of B is immaterial in this context, because the product of each row-vector of B' with each row-vector of B is zero, independent of other rows.

Algorithm: The corollary to Theorem 2 suggests an algorithm for finding out whether a given graph is self-dual. It is as follows: Write the incidence matrix A of the given graph with any order of its edges, say; $(e_1, e_2, \dots, e_m)$, where $m = 2(n-1)$. Now permute the columns of A to order $(e_{i_1}, e_{i_2}, \dots, e_{i_m})$ to get a new matrix A_i . Check to see if $A^t A_i = 0 \pmod{2}$. If $A^t A_i = 0$, graph G is self-dual. If $A^t A_i \neq 0$, find another matrix A_j and check $A^t A_j$. Continue this process till all permutations of $(e_1, e_2, \dots, e_m)$ are exhausted.

Obviously, for a graph of any appreciable size, this algorithm is impractical, because it takes too long to exhaust all permutations.

Theorem 3: For every $n \geq 4$, there exists at least one graph with n vertices which is self-dual. The degrees of vertices of this graph are $(n-1, 3, 3, 3, \dots, 3)$.

Proof: This theorem can be proved by induction on n . From Figs. 1, 2, and 3a, we know that the theorem is true for $n = 4, 5$, and 6 . Let us assume that there exists a self-dual, wheel-shaped graph for n vertices, as shown in Fig. 5a. By introducing an extra vertex (Fig. 5b) in one of the rim-segments and constructing its dual, we can show that the theorem is also good for $n+1$ vertices.

*The superscript t over a matrix stands for the transposed matrix.

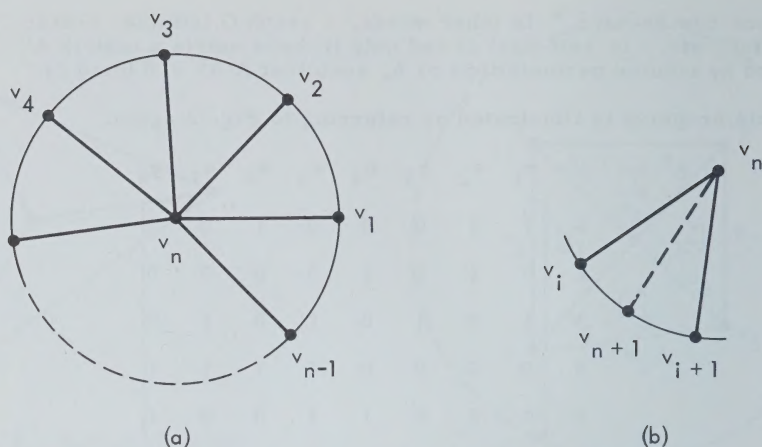


Fig. 5. A self-dual graph of n vertices

III. ENUMERATION OF SELF-DUAL GRAPHS

A very natural and interesting question can now be raised. What is the total number of self-dual graphs of n vertices?

Let us first consider the problem of constructing all graphs with n vertices and $2(n-1)$ edges. Since there are $n(n-1)/2$ possible pairs of vertices, if we label all vertices, we will get

$$\binom{\frac{n(n-1)}{2}}{2(n-1)}$$

graphs (column (a) of Table 1).

But many of these graphs are isomorphic. The number of distinct graphs with n (unlabeled) vertices and $2(n-1)$ edges is obtained from Pólya's Hauptsatz theorem [9, 10, 11]. This number is obtained as the coefficient of a generating function, which is produced by substituting polynomials $(1+x)$, $(1+x^2)$, $(1+x^3)$, $\dots$, etc., in the cycle-index of Pólya. For $n = 1, 2, \dots, 7$, these numbers are given in column (b) of Table 1.

Of all distinct graphs of n vertices and $2(n-1)$ edges, we are interested in the small subset of graphs that are (a) connected, (b) nonseparable, (c) planar, and (d) have no vertex of degree < 3 .

The number of distinct connected graphs has been obtained by Harary using Pólya's theorem, generalized by De Bruijn [11, 12, 13]. This number for $n = 1, 2, \dots, 7$ is given in column (c) of Table 1.

The problem of generating all planar graphs of n vertices has not been solved. Nor has the problem of generating all nonseparable graphs been solved [11]. In fact, if a generating function for the number of planar graphs can be found it might be an important step toward solving the four-color problem [12].

n	(a)	(b)	(c)	(d)
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	1	1	1	1
5	45	2	2	1
6	3,003	15	14	3
7	293,930	131	126	5

Table 1. Enumeration of graphs with n vertices

Therefore, although the number of planar, connected, nonseparable graphs of $2(n-1)$ edges, with no vertex of degree < 3 , will give an upper bound for the number of self-dual graphs, it does not seem the proper approach to take. Perhaps a direct attempt at using the generalized Pólya's theorem may be more fruitful [14]. Up to $n = 7$, the number of self-dual graphs is given in column (d) of Table 1.

IV. SELF-DUAL DIGRAPHS

A. Construction and Properties of Self-Dual Digraphs

A graph in which every edge has been assigned an orientation is called a directed graph or a digraph. Every planar, connected, and nonseparable digraph has a unique dual, which can be easily constructed geometrically [Ref. 2, pp. 45-48]. The dual is constructed first, disregarding the orientation of the edges, and then an orientation is assigned to each edge of the dual according to whether the corresponding edge in the original graph has the same direction as the circuit or the opposite. It can be easily shown that two digraphs are dual of each other if and only if the incidence matrix of one is a fundamental circuit matrix of the other [Ref. 3, p. 115].

Just as in case of nondirected graphs, only simple digraphs will be considered. The following theorem follows immediately from the considerations above.

Theorem 4: A necessary (but not sufficient) condition for a digraph to be self-dual is that the corresponding nondirected graph be self-dual.

That this condition is not sufficient can be readily seen by a counter example in Fig. 6. The nonoriented graph corresponding to these digraphs is self-dual. But these digraphs dual to each other are not isomorphic.

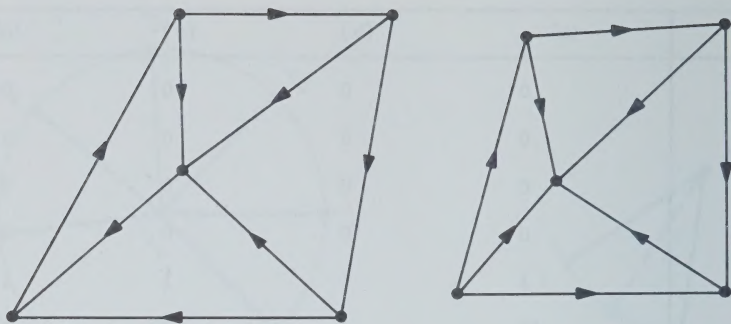


Fig. 6. Two dual digraphs (non-self-dual)

Corresponding to Theorems 1, 2, and 3 in Section II for nondirected graphs, we have the following three theorems for digraphs, which can be proved in the same way as Theorems 1, 2, and 3.

Theorem 5: The necessary (but not sufficient) conditions for a digraph G to be self-dual are

- (i) The number of edges in G is $e = 2(n-1)$, where n is the number of vertices in G .
- (ii) The sum of all in-degrees and out-degrees $d(v_i)_- + d(v_i)_+ \geq 3$ for every vertex v_i of G .

Theorem 6: A digraph is self-dual if and only if its incidence matrix is also its fundamental circuit matrix when the meshes are chosen as the fundamental circuits.

Theorem 7: For every $n \geq 4$, there exist at least two self-dual digraphs of n vertices.

B. Enumeration of Self-Dual Digraphs

From Theorem 4, we know that for a digraph to be self-dual, its corresponding nonoriented graph has to be self-dual. Corresponding to any given nonoriented self-dual graphs there are many different digraphs [Ref. 15, for exact numbers]. Some of these digraphs will be self-dual, while others will not be self-dual. For example, the two non-self-dual digraphs in Fig. 6 correspond to the self-dual nonoriented graph of Fig. 2. But the digraph in Fig. 7, corresponding to the self-dual nonoriented graph of Fig. 3(c), is a self-dual digraph. The determination of the number of distinct self-dual digraphs corresponding to a self-dual nonoriented graph is still an unsolved problem.

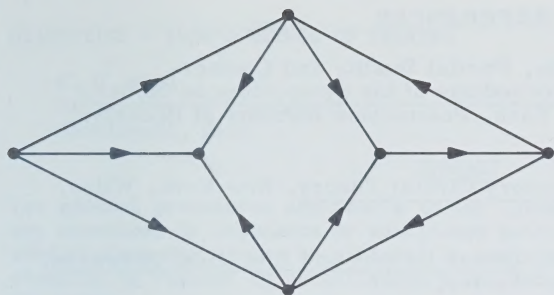


Fig. 7. A self-dual digraph corresponding to the self-dual graph of Fig. 3(c)

V. CONCLUSIONS AND FURTHER PROBLEMS

The nature of self-dual graphs and digraphs has been studied in this paper for the first time, in the knowledge of the author. Several interesting and important properties have been discovered, but many challenging problems remain unsolved. Some of these are as follows:

- (1) Although Theorem 2 provides an algorithm for detecting self-duality in a graph, the algorithm is very inefficient because it involves $n! - 1$ matrix multiplications. To find an algorithm where the number of steps increases linearly with n is still an open question.
- (2) Another unsolved problem is the enumeration of self-dual graphs for a given n . It does not seem probable that this number can be found in a closed form, but there must be a generating function corresponding to it.
- (3) What is the number of self-dual digraphs and non-self-dual digraphs corresponding to a given nonoriented self-dual graph? Is it true that for every nonoriented self-dual graph there is at least one self-dual digraph?
- (4) Graphs that are not totally self-dual but only partially self-dual can also be studied. Partial self-duality can be defined appropriately.

VI. ACKNOWLEDGMENTS

The author expresses his appreciation to Professor S. L. Hakimi of Northwestern University and Paul Sobel of the Jet Propulsion Laboratory (JPL) for their valuable suggestions. He is also thankful to J. J. Wedel of JPL for encouragement and help.

REFERENCES

1. E. Colin Cherry, "Duality, Partial Duality and Contact-Transformations," in Proceedings of the Symposium on Modern Network Synthesis, New York, Polytechnic Institute of Brooklyn, pp. 323-347, Apr. 1955.
2. E. A. Guillemin, Introductory Circuit Theory, New York, Wiley, 1953.
3. S. Seshu and M. B. Reed, Linear Graphs and Electrical Networks, Reading, Mass., Addison-Wesley, 1961.
4. H. S. M. Coxeter, "Self-dual Configurations and Regular Graphs," Bull. Am. Math. Soc., Vol. 56, pp. 413-455, Sept. 1950.
5. R. Artzy, "Self-dual Configurations and Their Levi Graphs," Proc. Am. Math. Soc., Vol. 7, pp. 299-303, Apr. 1956.
6. C. Berge, The Theory of Graphs, New York, Wiley, 1962.
7. O. Ore, Graphs and Their Uses, New York, Random House, 1963.
8. R. G. Busacker and T. L. Saaty, Finite Graphs and Networks, New York, McGraw-Hill, 1965.
9. G. Pólya, "Kombinatorische Anzahlbestimmungen für Gruppen, Graphen, und chemische Verbindungen," Acta Math., Vol. 68, pp. 145-254, 1937.
10. J. Riordan, An Introduction to Combinatorial Analysis, New York, Wiley, 1958.
11. F. Harary, "The Number of Linear, Directed, Rooted and Connected Graphs," Trans. Am. Math. Soc., Vol. 78, pp. 445-463, 1955.
12. F. Harary, "Combinational Problems in Graphical Enumeration," in Applied Combinational Mathematics, ed. E. F. Beckenback, New York, Wiley, pp. 185-212, 1964.
13. N. G. DeBruijn, "Generalization of Pólya's Fundamental Theorem in Enumerative Combinational Analysis," Indagationes Mathematicae, Vol. 21, pp. 59-69, 1959.
14. G. W. Ford and G. E. Uhlenbeck, "Combinational Problems in the Theory of Graphs: I., II., III., IV.," Proc. Nat. Acad. Sci. U.S.A., Vol. 42, 1956, pp. 122-128, 203-208, 529-535; Vol. 43, pp. 163-167, 1957.
15. F. Harary, "The Number of Oriented Graphs," Mich. Math. J., Vol. 4, pp. 221-224, 1957.

K. U. WANG
University of Cincinnati
Cincinnati, Ohio

SUMMARY

Two general procedures applicable to any linear systems whose components are described by impedance or admittance matrices, are derived; the torn subdivisions may either be attached or not attached. Thus a better understanding is reached by establishing a unified mathematical theory for diakoptics.

INTRODUCTION

By defining the vector spaces and the dual vector spaces associated with the closed-paths, the open-paths and the edges of a directed linear graph, we can establish four exact sequences. These exact sequences contain the linear transformations which yield Kron's nonsingular transformation matrices, orthogonal network equations and transformations for interconnections and tearing. It can be shown that the classical linear graph theory of electrical networks is encompassed within this development; for instance, the branch and chord formulations are merely special cases of Kron's orthogonal networks.

A large-scale linear system and its linear graph can be torn into s subdivisions (disconnected subgraphs) and t tie linear (disconnected edges). To maintain the invariance of component voltages and currents in the torn network, we can assume that new, unknown current or voltage sources are introduced at the points of tearing. For the ease of computation, the subdivisions are not to have mutual couplings of any kind, nor is coupling between any one subdivision and the tie lines allowed. The methods used in diakoptics will be classified according to the terminal representation chosen for the subdivision, and the subdivisions can be either attached or not attached.

ALL-OPEN METHOD

This method presumes that the components are described by an admittance matrix. Let the subdivisions be not attached. By solving the orthogonal network equation for the open-path current vector, each subdivision is represented by an all-open network (an $n-1$ branch Lagrangian tree). Let the collection of these trees and the tie lines be denoted by K_{j1} and K_{j2} respectively. The interconnected graph K_k is formed by restoring tie lines K_{j2} to the proper vertices. The nonsingular transformation matrix T_{jk} relating the torn network K_j and the interconnected network K_k is

$$T_{jk} = \frac{j1}{j2} \begin{bmatrix} \begin{matrix} o1 & o2 & c \end{matrix} \\ \begin{matrix} U & T_1 & T_2 \\ 0 & T_3 & T_4 \end{matrix} \end{bmatrix},$$

where "ol" and "jl" denote the same set of open-paths, "o2" is the new open-paths of K_k , and "c" is the closed-path in K_k . The impedance matrix Z_k of K_k is obtained from

$$Z_k = T'_{jk} Z_j T_{jk},$$

where Z_j is obtained by inverting $s+1$ diagonal submatrices of the admittance matrix of K_j . The closed-path current vector of K_k is obtained by inverting the submatrix of Z_k corresponding to the closed-paths "c". Then, the open-path voltage vector of the given network is found from Z_k . Thus only the inversions of $s+2$ small matrices are required for the solution. For the case of subdivisions touching at one vertex, the transformation matrix becomes

$$T_{jk} = \frac{j_1}{j_2} \begin{bmatrix} o & c \\ -\frac{U}{0} + \frac{T}{U} \end{bmatrix} .$$

ALL-CLOSED METHOD

Assume that the components are described by an impedance matrix. By solving the orthogonal network equation for the closed-path voltage vector, each subdivision is represented by a terminal graph composed of self-loops. Let the "s" terminal graphs and the tie lines be denoted by K_{j1} and K_{j2} respectively. The interconnected network K_k is obtained by restoring the tie lines, the transformation matrix $(T_{jk})^{-1}$ is found to be

$$(T_{jk})^{-1} = \frac{j_2}{j_1} \begin{bmatrix} o & c \\ -\frac{U}{T} + \frac{0}{U} \end{bmatrix} .$$

The admittance matrix Y_k of K_k is obtained from

$$Y_k = (T_{jk})^{-1} Y_j (T_{jk})^{-1} ,$$

where Y_j consists of $s+1$ diagonal submatrices. The open-path voltage vector of K_k is found by inverting the submatrix of Y_k corresponding to the open-paths. Then, the closed-path current vector of the given network can be obtained.

In conclusion, this paper presents a unified approach to diakoptics, and its treatment on Kron's Poisson-type networks (all open-method with subdivisions not attached) is not found elsewhere. Finally, the author wishes to thank Dr. Russell Dunholter for his guidance and encouragement.

REFERENCES

1. Kron, G., Diakoptics-The Piecewise Solution of Large Scale Systems. Macdonald. London, 1963.
2. Nickerson, H.K., Spencer, D.C., and Steanrod, N.E., Advanced Calculus. D. Van Nostrand. New York, 1959.
3. Dunholter, R. and Wang, K.U., "The One-network Theory and Diakoptics." Proceeding of the 11th Midwest Symposium on Circuit Theory, May 13-14, 1968.
4. Wang, K.U., "The Mathematical Foundation of Kron's Tensor Analysis of Electrical Networks." PhD Dissertation, University of Cincinnati, 1967.

ALGEBRAIC GENERATION OF NETWORK FUNCTION COEFFICIENTS AND PARTIAL DERIVATIVES

GERRY L. KELLY
Department of Electrical Engineering
University of Kansas
Lawrence, Kansas

ABSTRACT

The primary purpose of this brief paper is to show that the topological quantities for a network function or the partial derivatives of a network function with respect to the network parameters can be algebraically manipulated to produce a computer generated Fortran subroutine. A subroutine which has been generated can then be used in the optimization of electrical networks or merely used for network analysis. This means that it is necessary to analyze a network for a given network topology only one time provided that the compiled version of the computer generated Fortran subroutine can be stored onto magnetic tape, cards, or other suitable storage.

INTRODUCTION

It is extremely important in the optimization of electrical networks to have an efficient method for obtaining the network function and at least the first partial derivatives of the network function with respect to the network parameters. The topological relationships for network functions are given in Seshu and Reed (1) and the topological relationships for the first and second partial derivatives of a network function with respect to the network parameters have been obtained by Kelly (2). By utilizing topological methods one can obtain closed form solutions for the network function coefficients and their partial derivatives; however, it is usually quite time consuming on a digital computer to obtain the necessary topological relationships each time an evaluation is required for a particular set of elements values. The method presented here utilizes a computer algorithm to find the required k-tree admittance products and then a computer generated Fortran subroutine is produced which is capable of evaluating the network function coefficients and the corresponding partial derivatives for a given set of element values. Thus for any given network topology it is necessary to perform an analysis only one time.

DISCUSSION

It is well known that a network transfer function, $F(s, x_1, x_2, \dots, x_n)$, for a n-element time invariant LLF network is bilinear in any x_i since any k-tree can contain a particular element at most once. It is clear that

$$F(s, x_1, x_2, \dots, x_n) = \frac{A(1) + A(2)s + \dots + A(r+1)s^r}{B(1) + B(2)s + \dots + B(m+1)s^m} \quad (1)$$

each $A(k)$ or $B(k)$ in equation (1) is a function of the network parameters. For a given network it is this functional dependence that we would like to algebraically generate in the form of a Fortran subroutine. By taking the partial derivative of equation (1) with respect to a network parameter x_j we obtain

This work was partially supported by Collins Radio Company, Newport Beach, California and by the National Science Foundation under Grant GK-3657.

$$\frac{\partial F(x_1, x_2, \dots, x_n)}{\partial x_j} = \frac{1}{B(s)} \left\{ \sum_{k=1}^{r+1} \frac{\partial A(k)}{\partial x_j} s^{k-1} - F(s) \sum_{k=1}^{m+1} \frac{\partial B(k)}{\partial x_j} s^{k-1} \right\} \quad (2)$$

By inspection of equation (2) it is evident that the coefficients and the first partial derivatives of the coefficients of $F(s, x_1, x_2, \dots, x_n)$ completely determine the first partial derivative of the network function with respect to each network parameter. It is easy to verify that the partial derivatives of the magnitude squared function and the delay of a network transfer function are also obtainable from the coefficients and the corresponding partial derivatives of the coefficients.

A three stage negative feedback amplifier is shown in Figure 1. The computer generated Fortran statements for the coefficients are shown in Figure 2. The reader is referred to Kelly (3) for the computer generated first and second partial derivatives of this example as well as more appropriate examples.

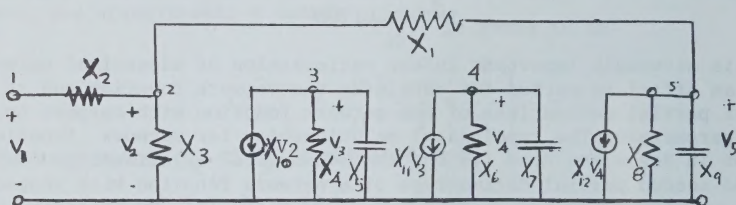


Figure 1. A Three Stage Negative Feedback Amplifier Equivalent Circuit

C FORTRAN STATEMENTS GENERATED FOR COEFFICIENTS

```
A(01)=+X(01)*X(02)*X(04)*X(06)-X(02)*X(10)*X(11)*X(12)
A(02)=+X(01)*X(02)*X(04)*X(07)+X(01)*X(02)*X(05)*X(06)
A(03)=+X(01)*X(02)*X(05)*X(07)
B(01)=+X(01)*X(02)*X(04)*X(06)+X(02)*X(03)*X(04)*X(06)
B(02)=+X(01)*X(02)*X(04)*X(07)+X(01)*X(02)*X(05)*X(06)+X(02)*X(03)
1)*X(04)*X(07)+X(02)*X(03)*X(05)*X(06)
B(03)=+X(01)*X(02)*X(05)*X(07)+X(02)*X(03)*X(05)*X(07)
```

Figure 2. Computer Generated Fortran Statements For Voltage Transfer Function

REFERENCES

1. S. Seshu and M.B. Reed, Linear Graphs and Electrical Networks, Reading, Mass., Addison-Wesley, 1961.
2. G. L. Kelly, "Nonlinear Programming and Multiparameter Sensitivity in Linear Time-Invariant Networks," Coordinated Science Laboratory Report No. 301, University of Illinois, 1966.
3. G. L. Kelly, "Computer Generated Fortran Subroutines for Network Function Coefficients and Associated Partial Derivatives," Center for Research in Engineering Science Technical Report No. 145-1, University of Kansas, 1968.

THE EFFECT OF STRAY PARAMETERS ON THE STABILITY OF A CLASS OF NON-LINEAR NETWORKS

S. T. SATHE and T. E. STERN
Columbia University, New York, N.Y.

Abstract: When analysing a nonlinear resistive network one normally considers its simplified resistive model which neglects the stray inductances and capacitances present in the network. The dynamics introduced by these stray elements may, in certain cases, critically affect the network response. For a certain class of resistive networks we show here that a certain lower bound on the ratio of the smallest stray capacitance to the largest stray loop inductance is sufficient for the global stability of the network.

I. INTRODUCTION

An important class of networks in current technology consists of those which are nonlinear and essentially resistive in nature. By this we mean that any reactive parameters occur unintentionally as stray or parasitic elements. Typical of these types of networks are those used in digital logic circuitry. For analysis purposes networks which are essentially resistive are normally approximated by resistive mathematical models, that is, models containing no dynamics. The analysis of such a model reduces to the determination of one or more equilibrium states. The physical network, however, contains certain dynamic effects due to the unavoidable presence of various stray reactive parameters, and these dynamic effects can lead to certain undesirable unstable behavior such as sustained parasitic oscillations. It is therefore of considerable practical importance to inquire into the conditions under which the response of an essentially resistive network from any initial state will, in spite of the stray reactances, remain bounded and tend toward an equilibrium state as $t \rightarrow \infty$. Under these circumstances the network will be said to possess a constant limiting regime (CLR). (A CLR implies the non-existence of any sustained oscillations).

The purpose of this paper is to delineate a class of essentially resistive networks in which sufficient conditions for a CLR can be derived and simply interpreted.

II. NETWORK DESCRIPTION

We shall limit the discussion to networks composed of elements whose resistive mathematical model satisfies the following conditions:

- 1) Each element has terminal relations which can be expressed in the form $i = f(v)$ where i , f and v are all n -vectors (column matrices) and f is continuously differentiable. Thus f represents a resistive n -port.
- 2) The function f is reciprocal [5] for each element.

3) For each element there exist positive constants N and ϵ such that* $\|v\| \geq N$ implies $v^t f(v) \geq \epsilon$

For any network element that is quasilinear (QL) [5], its terminal relations can also be written in the inverse form $v = f^{-1}(i)$. All elements that are not QL will be called voltage-controlled nonlinear resistors (VCNR). Note that any form of (reciprocal) nonlinear coupling is permitted among the various ports of an element. As will be apparent from our development, the dual case of networks containing QL resistors and current-controlled nonlinear resistors (CCNR) can be handled in an identical manner.

III. PROPER AUGMENTATION, NETWORK EQUATIONS

Since the actual distribution of stray reactive elements is not easy to determine, especially when the physical layout of the elements is unknown, we would like to obtain our results with as little information as possible concerning these parameters. The resistive network model will therefore be augmented by certain linear capacitances and inductances in a largely arbitrary fashion, subject to the following considerations.

It is known that for certain "irregular" nonlinear networks [1] the response of the mathematical model may be indeterminate. To avoid this possibility, we shall assume that each VCNR element has a positive capacitance of arbitrary value in parallel with each port. For the QL elements we shall assume that each port is either augmented by a series positive inductance or by a parallel positive capacitance (all of arbitrary values). (For simplicity, it will be assumed that all ports of the same element are augmented by the same type of reactance). Any augmentation consistent with the above restrictions will be said to be a "proper augmentation". In the sequel it will always be assumed that we are dealing with a properly augmented network.

To obtain the state variable equations for such a network we choose a "proper tree" following the procedure of Bryant [3] and write the fundamental loop and cutset matrices, following the notation of [5].

$$B = [I \ F] = \begin{bmatrix} I_{cc} & 0 & 0 & F_{cs} & 0 & 0 \\ 0 & I_{gg} & 0 & F_{gs} & F_{gr} & 0 \\ 0 & 0 & I_{\gamma\gamma} & F_{\gamma s} & F_{\gamma r} & F_{\gamma \ell} \end{bmatrix}$$

$$\text{and } Q = [-F^t \ I]$$

where $F_{gr} = 0$ because in the augmented network each resistive element has a capacitance in shunt or an inductance in series with each of its ports.

* $\|\cdot\|$ denotes "norm" and $(\cdot)^t$ denotes "transpose".

The network differential equations will be [2]

$$\begin{bmatrix} \dot{\lambda} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} \dot{i}_Y \\ \dot{e}_s \end{bmatrix} = \begin{bmatrix} -F_{Yr} e_r - F_{Ys} e_s \\ +F_{gs}^t i_g + F_{Ys}^t i_Y \end{bmatrix} \quad (1)$$

where $L = L_Y + F_{Y\ell} L_\ell F_{Y\ell}^t$

and $C = C_s + F_{cs}^t C_c F_{cs}$ are constant symmetric positive definite matrices, $i_g = g(e_g)$ is the current vector for the resistive elements included in the chords and e_r is the voltage vector for the resistive elements in the tree branches. Since only QL resistive elements may appear as tree branches, we can write $e_r = f^{-1}(i_r) = h(i_r)$.

The form of these equations is greatly simplified if we define a scalar "dissipation function"

$$\mathfrak{F}(e_s, i_Y) = -\int g^t(-F_{gs} e_s) F_{gs} de_s - \int h^t(F_{Yr}^t i_Y) F_{Yr}^t di_Y - i_Y^t F_{Ys} e_s \quad (2)$$

where $\mathfrak{F}$ is assumed to be an indefinite line integral, (independent of path because of reciprocity [5]).

The network equations now become

$$\begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} \dot{i}_Y \\ \dot{e}_s \end{bmatrix} = \begin{bmatrix} (\partial \mathfrak{F} / \partial i_Y)^t \\ -(\partial \mathfrak{F} / \partial e_s)^t \end{bmatrix} \quad (3)$$

IV. EXISTENCE OF A CONSTANT LIMITING REGIME

Based on Eq. (3) we can now derive a sufficient condition for the existence of a CLR. The following definitions will be required.

Let $R(i_Y) \triangleq F_{Yr} \frac{\partial h}{\partial i_r} F_{Yr}^t$

$G(e_g) \triangleq F_{gs}^t \frac{\partial g}{\partial e_g} F_{gs}$

If A is a real symmetric matrix, let

$$\begin{aligned} m(A) &= \text{least eigenvalue of } A \\ M(A) &= \text{greatest eigenvalue of } A \end{aligned}$$

$$\text{Let } m_R = \inf m(R) \text{ over all } i_R \quad (4a)$$

$$m_G = \inf m(G) \text{ over all } e_G \quad (4b)$$

$$m_C = m(C) \quad (4c)$$

$$M_L = \frac{1}{2} M(L) \quad (4d)$$

Now, consider a properly augmented network N, described by equations (3) and satisfying the assumptions of Sect. II. In terms of the above quantities the following theorem may be stated:

Theorem: The network N will possess a constant limiting regime if

$$\frac{m_C}{M_L} > - \frac{2m_G}{m_R} \quad (5)$$

Before proceeding to the proof of the theorem, it is worthwhile to give a simple interpretation of condition (5) in terms of the structure of the network. It is easily shown that m_C is equal to or greater than the smallest capacitor in N, and M_L is equal to or less than the largest loop inductance in N, (i.e., the sum of all inductances in the loop [7]). Furthermore, m_R is derived from the incremental resistance matrix seen from the inductors when all capacitors are short-circuited. Because of the manner of augmentation, m_R must be positive.* On the other hand, m_G is related to the incremental conductance matrix seen from the capacitors (with inductors open-circuited), and may be negative.

Clearly if m_G were positive, condition (5) would automatically be fulfilled for any proper augmentation. In the more interesting case where m_G is negative the theorem indicates that the ratio of smallest capacitance to largest loop inductance must be above a certain minimum value, $\frac{-2m_G}{m_R}$, in order to assure a CLR. (This is, of course, a sufficient condition only).

Proof: We prove the theorem using two Lyapunov functions,

$$W = \frac{1}{2} e_s^t C e_s + \frac{1}{2} i_Y^t L i_Y \quad (6)$$

$$\text{and } V = \frac{a\mathfrak{F}}{2} + \frac{1}{2} \left[\frac{\partial \mathfrak{F}}{\partial (i_Y, e_s)} \right]^t \begin{bmatrix} L^{-1} & 0 \\ 0 & C^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathfrak{F}}{\partial (i_Y, e_s)} \end{bmatrix} \quad (7)$$

* See proof of theorem.

The latter was first suggested by Brayton and Moser [6]. In (7), a is a positive constant to be determined subsequently. Using the direct method of Lyapunov [4], we can show the existence of a CLR if

1) $\dot{V} < 0$ everywhere except at the singular points of (3).

2) All trajectories are bounded for $t > 0$. [8, p. 66]

To prove condition (1), we take

$$\begin{aligned} \dot{V} &= \begin{bmatrix} \dot{i}_Y \\ \dot{e}_s \end{bmatrix}^t \left\{ \frac{a}{2} \begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix} + \begin{bmatrix} -R & F_{Ys} \\ -F_{Ys}^t & -G \end{bmatrix} \right\} \begin{bmatrix} i_Y \\ e_s \end{bmatrix} \\ &= \dot{i}_Y^t \left(-R + \frac{a}{2} L \right) i_Y + \dot{e}_s^t \left(-G - \frac{a}{2} C \right) e_s \end{aligned} \quad (8)$$

Since each VCNr is shunted by a capacitance, it must be a chord of a proper tree. Thus, all resistive tree branches are necessarily QL. Hence the function $h(i_r)$ must be QL in i_r . Since each inductive chord must be in series with a resistive element, the rank of the matrix F_{Yr} will be equal to the number of rows of F_{Yr} . As a result the function $F_{Yr}h(F_{Yr}i_r)$ will be QL in i_Y and the matrix R will be symmetric positive definite. [5, p. 36] No such conclusions can be drawn concerning the matrix G .

Now using definitions (4a-d) and substituting into (8) we may write

$$\dot{V} \leq \dot{i}_Y^t \left(aM_L - m_R \right) i_Y + \dot{e}_s^t \left(-m_G - \frac{am_C}{2} \right) e_s$$

Assuming that condition (5) is fulfilled we can find an $a > 0$ satisfying,

$$\frac{m_R}{M_L} > a > \frac{-2m_G}{m_C}$$

Such a value of a will make $\dot{V} \leq 0$ and $\dot{V} = 0$ only when $\dot{i}_Y = 0$ and $\dot{e}_s = 0$. Thus condition (1) is fulfilled.

To prove condition (2) we find from matrix B and eq. (1) that

$$\dot{W} = -e_g^t g(e_g) - i_r^t h(i_r)$$

It was assumed that $e_g^t g(e_g) \geq \epsilon > 0$ for large $\|e_g\|$ and $h(i_r)$ is a quasilinear function. Hence $g_W \rightarrow \infty$ and $\dot{W} < 0$ when $\|e_g\| + \|i_r\| \rightarrow \infty$. Due to the method of augmentation $\|i_v\| + \|e_s\| \rightarrow \infty$ always implies $\|e_g\| + \|i_r\| \rightarrow \infty$. Thus using W as the Lyapunov function we can conclude that all solutions of eq. (1) will remain bounded.

This completes the proof of the theorem.

V. CONCLUSIONS

We have shown that by assuming physically reasonable distributions of stray reactive elements in a basically resistive network, certain sufficient conditions for a stable behavior can be derived. For a network composed of VCNR and QL resistive elements the conclusion to be drawn from these conditions is that stability will be assured if the stray inductances are sufficiently small compared to the stray capacitances. As a practical application of this result, consider the following situation. Suppose a network of the type discussed herein is constructed and is found to oscillate due to parasitic effects. Our results suggest that such oscillations can be suppressed by either reducing any stray inductances, or adding capacitances. (The latter may be easier to implement).

For the sake of clarity and brevity, many simplifications were made in the above development. Considerably sharper and more general results can be obtained by omitting these simplifications. For example, one can show that it is the smallest capacitance in parallel with a VCNR that determines the limit of stability, rather than the smallest capacitance in the network. Thus, to stabilize a network one should increase all VCNR capacitances.

Finally, we recall that all of the results can be dualized. Thus, for example, a network composed of CCNR's and QL resistances will be stable if its stray capacitances are sufficiently small compared to its stray inductances.

REFERENCES

1. On the Analysis of Nonlinear Irregular Networks - Oh, Stern, and Meadows; Symposium on Generalized Networks, Poly. Tech. Inst. of Brooklyn, 1966.
2. On the Equations of Nonlinear Networks - T. E. Stern, I.E.E.E. Trans. on Circuit Theory, Vol. 13, No. 1, March 1966.
3. The Order of Complexity of Electrical Networks - P. R. Bryant, I.E.E.E. (London) Monograph 335E, June 1959.
4. Probleme General de la Stabilite du Mouvement - M. A. Lyapunov.
5. Theory of Nonlinear Networks and Systems - T. E. Stern, Addison-Wesley, 1965.
6. Theory of Nonlinear Networks I, II - R. K. Brayton and J. K. Moser, Quarterly of Applied Mathematics, Vol. 22, April and July 1964.
7. Uber die Abgrenzung der Eigenwerte einer Matrix - S. Gersgorin Izv. Akad. Nauk, SSSR, 7, 749-754, (1931).
8. Stability by Lyapunov's Direct Method with Applications - J. LaSalle and S. Lefschetz.

L^p -STABILITY ($1 \leq p \leq \infty$) OF NONLINEAR TIME-VARYING FEEDBACK SYSTEMS*

M. Y. WU and C. A. DESOER

Department of Electrical Engineering and Computer Sciences and the
Electronics Research Laboratory, University of California, Berkeley

I. INTRODUCTION

In the past few years the L^2 -stability [1 - 2] and the L^∞ -stability [3 - 5] of certain classes of nonlinear and time-varying, feedback systems have been extensively studied. Recently Desoer and Wu [6 - 8] have derived L^p -stability conditions for a very broad class of linear time-invariant systems whose impulse responses may include an integrator and an infinite sequence of impulses. Chen [9] has considered the L^p -stability for a class of linear time-varying systems. In this paper the transformation technique, the small gain theorem and some results of Desoer and Wu [6 - 8] are used to derive the L^p -stability for a class of nonlinear time-varying systems. As an application, L^p -stability conditions for the damped Mathieu equation are obtained and compared with recent zero-input results.

II. NOTATIONS

In the following, we shall encounter real numbers, vectors (in R^n) and elements of function spaces. Lower case boldface (e.g., $\underline{e}$, $\underline{u}$) denotes vectors and upper case boldface (e.g., $\underline{K}$, $\underline{G}$) denotes matrices; R_+ denotes the set of non-negative real numbers. The symbol $|\cdot|$ is used to denote both the magnitude of a real number and the norm of a vector in R^n . For function spaces, we use the following norms: let $\underline{x}: R_+ \rightarrow R^n$, then, by definition,

$$\|\underline{x}\|_p \triangleq \left[\int_0^\infty |x(t)|^p dt \right]^{1/p} \quad (1 \leq p < \infty)$$

and

$$\|\underline{x}\|_\infty \triangleq \sup_{t \geq 0} |x(t)| < \infty \quad (p = \infty)$$

The resulting normed spaces are denoted by L^n_p ($1 \leq p \leq \infty$). If $n = 1$ (scalar case), we write L^p . When the symbol $|\cdot|$ and $\|\cdot\|$ are applied to a matrix or a matrix-valued function, they denote the induced operator norms. Note that the norms defined above are valid independently of the choice of norm in R^n because all norms in R^n are equivalent.

*Research sponsored by the National Aeronautics and Space Administration under Grant Nsg-354(Suppl. 5) and the National Science Foundation under Grant GK-2277.

Following Zames [2], the space L_{ne}^p , the extension of L_n^p space, is defined as follows:

$$L_{ne}^p \triangleq \{ \tilde{x}(\cdot) \mid \int_0^T |x(t)|^p dt < \infty \quad \forall T \in [0, \infty), \quad 1 \leq p < \infty \}$$

and

$$L_{ne}^\infty \triangleq \{ \tilde{x}(\cdot) \mid \sup_{t \in [0, T]} |x(t)| < \infty, \quad \forall T \in [0, \infty) \}$$

Roughly speaking, if $\tilde{x} \in L_{ne}^\infty$, then $\tilde{x}$ does not have a finite escape time.

In order to allow us to consider a larger class of linear subsystems whose impulse responses may include an infinite sequence of impulses, we introduce the Banach algebra $\mathcal{A}_n$ [6 - 8]: Let $\tilde{f}$ be a distribution whose support is in $[0, \infty)$. We say that $\tilde{f}$ is an element of $\mathcal{A}_n$ if

$$\tilde{f}(t) = \tilde{f}_a(t) + \sum_{i=0}^{\infty} \tilde{f}_i \delta(t - t_i)$$

where $\tilde{f}_a: [0, \infty) \rightarrow R^n$ is in L_n^1 , the sequence $\{t_i\}_0^\infty$ is in $[0, \infty)$ with $0 = t_0 < t_1 < t_2 < \dots$; $\{\tilde{f}_i\}$ is a sequence of constant vectors in R^n

subject to $\sum_{i=0}^{\infty} |\tilde{f}_i| < \infty$, and δ is the Dirac "function". The set of all

elements in $\mathcal{A}_n$ constitutes a commutative Banach algebra with the usual definition for addition, the product defined by convolution, and the norm defined by

$$\|\tilde{f}\| \triangleq \int_0^\infty |\tilde{f}_a(t)| dt + \sum_{i=0}^{\infty} |\tilde{f}_i|$$

These facts are well known [10 - 11]. Similarly, we say that an $n \times n$ matrix-valued distribution $\tilde{F}$ is in $\mathcal{A}_n \times \mathcal{A}_n$ whenever each of its column vectors is in $\mathcal{A}_n$. If $n = 1$, we write $\mathcal{A}$.

The symbol " $\hat{\cdot}$ " over a function, such as $\hat{\tilde{f}}$, denotes the Laplace transform of $\tilde{f}$; it is defined by

$$\hat{\tilde{f}}(s) = \int_0^\infty \tilde{f}(t) e^{-st} dt$$

For distributions, it is defined according to L. Schwartz [15] or, by using Stieltjes integrals, according to Widder [16].

The subscript T , as in $\tilde{f}_T$, denotes the truncation of the function $\tilde{f}$ at time T , namely,

$$\tilde{f}_T(t) = \begin{cases} \tilde{f}(t) & \text{for } 0 \leq t \leq T \\ 0 & \text{for } t > T \end{cases}$$

III. SYSTEM DESCRIPTIONS

In this paper we shall consider the multiple-input, multiple-output, nonlinear, time-varying system $\tilde{S}$ as shown in Fig. 1. The vectors $\underline{u}_1(t)$, $\underline{u}_2(t)$, $\underline{e}_1(t)$, $\underline{e}_2(t)$, $\underline{x}_1(t)$ and $\underline{y}_2(t)$ belong to R^n . The block labeled $\tilde{G}$ is a linear, time-invariant, nonanticipative subsystem whose input-output relation is defined in terms of its impulse response matrix $\tilde{G}$ by the convolution integral

$$\underline{x}_1(t) \triangleq \begin{cases} (\tilde{G} * \underline{e}_1)(t) = \int_0^t \tilde{G}(t - \tau) \underline{e}_1(\tau) d\tau & \text{for } t \geq 0 \\ 0 & \text{for } t < 0 \end{cases} \quad (1)$$

The block labeled $\tilde{\Phi}_t$ is a memoryless, time-varying nonlinearity whose input-output relation is defined in terms of a nonlinear function

$\varphi: R^n \times R_+ \rightarrow R^n$ by

$$\underline{y}_2(t) = \varphi[\underline{e}_2(t), t] \quad (2)$$

We assume throughout that the inputs (which also take into account the effects of initial conditions and outside disturbances) $\underline{u}_1(t)$, $\underline{u}_2(t)$ are zero for $t < 0$. The system equations (See Fig. 1) are Eqs. (1), (2) and

$$\underline{e}_1 = \underline{u}_1 - \underline{y}_2 \quad (3)$$

$$\underline{e}_2 = \underline{u}_2 + \underline{y}_1 \quad (4)$$

IV. MAIN RESULTS

Theorem 1.

Consider the system $\tilde{S}$ (Fig. 1). Let the input-output relation of the linear time-invariant subsystem $\tilde{G}$ be given by (1), where the open-loop impulse response matrix $\tilde{G}$ is of the form

$$\tilde{G}(t) = \begin{cases} R + \tilde{G}_t(t) & \text{for } t \geq 0 \\ 0 & \text{for } t < 0 \end{cases} \quad (5)$$

with R being an $n \times n$ constant real matrix and $\tilde{G}_t \in Q_{n \times n}$. Let $\tilde{\Phi}_t$ be the time-varying nonlinearity whose characteristic $\varphi(\cdot, \cdot)$ has the following properties:

N1. $\varphi(\cdot, \cdot): R^n \times R_+ \rightarrow R^n$ and φ is a continuous function with respect to its first argument and is a regulated function [12] with respect to its second argument.

N2. There is an $n \times n$ constant real matrix $\tilde{K}$ and a positive real number λ such that

$$|\varphi(\sigma, t) - \tilde{K} \sigma| \leq \lambda |\sigma| \quad \forall t \in R_+, \forall \sigma \in R^n \quad (6)$$

Let, for some p in $[1, \infty]$, the inputs u_1, u_2 be in L_n^p and, for all such inputs, the corresponding e_1, e_2 be in L_{ne}^p . Under these conditions if

$$(a) \quad \inf_{\operatorname{Re} s \geq 0} |\det[\tilde{I} + \hat{G}(s) \tilde{K}]| > 0 \quad (7)$$

and if either $R = 0$ or all the eigenvalues of $R \tilde{K}$ are in the open right half plane,

$$(b) \quad \|\tilde{H}_K\| \lambda < 1$$

$$\text{where } \tilde{H}_K(t) \triangleq \mathcal{L}^{-1}\{[\tilde{I} + \hat{G}(s) \tilde{K}]^{-1} \hat{G}(s)\} \quad (9)$$

then e_1, e_2, y_1 and y_2 are in L_n^p .

As a special case of Theorem 1, we state the results for the scalar case as Corollary 1.

Corollary 1.

Consider the system S (Fig. 1). Let the input-output relation of the linear time-invariant subsystem G be defined in terms of its open-loop impulse response g by

$$y_1(t) = \begin{cases} (g * e_1)(t) & \text{for } t \geq 0 \\ 0 & \text{for } t < 0 \end{cases} \quad (10)$$

where g is of the form

$$g(t) = \begin{cases} r + g_\ell(t) & \text{for } t \geq 0 \\ 0 & \text{for } t < 0 \end{cases} \quad (11)$$

with r being a non-negative constant and $g_\ell \in \mathcal{Q}$. Let φ_t be the time-varying nonlinearity whose characteristic $\varphi(\cdot, \cdot)$ has the following properties:

N1. $\varphi(\cdot, \cdot): R \times R_+ \rightarrow R$ and φ is a continuous function with respect to its first argument and is a regulated function with respect to its second argument.

N2. For some finite constants k_1 and k_2

$$0 < k_1 \sigma^2 \leq \sigma \varphi(\sigma, t) \leq k_2 \sigma^2 \quad \forall t \in R_+, \forall \sigma \in R \quad (12)$$

Let, for some p in $[1, \infty]$, the inputs u_1, u_2 be in L^p and, for all such inputs, the corresponding e_1, e_2 be in L_e^p . Under these conditions, if for some constant $k \in [k_1, k_2]$

$$(i) \quad \inf_{\operatorname{Re} s \geq 0} |1 + k \hat{g}(s)| > 0 \quad (13)$$

$$(ii) \quad \|h_k\| \lambda < 1 \quad (14)$$

where

$$h_k(t) \triangleq \mathcal{L}^{-1}\{[1 + k \hat{g}(s)]^{-1} \hat{g}(s)\} \quad (15)$$

and

$$\lambda \triangleq \max\{|k_2 - k|, |k_1 - k|\} \quad (16)$$

then e_1 , e_2 , y_1 and y_2 are in L^p .

Comments.

1. If, in Corollary 1, we take $k = (k_1 + k_2)/2$, then N2 may be written as $|\varphi(\sigma, t) - k \sigma| \leq \lambda |\sigma|$, with $\lambda = |k_2 - k_1|/2$, which is the direct specialization of N2 to the scalar case.

2. In Corollary 1, since both h_k and λ depend on k , there may exist an optimal choice of k such that Eq. (14) will be satisfied for the largest class of the linear subsystems.

3. If, in Eq. (8), r is zero, then the assumption N2 can be relaxed in that k_1 in N2 need not be positive.

4. If the assumptions of Corollary 1 are specialized to $g \in L^1$ and $p = \infty$, then Corollary 1 reduces to a result of Sandberg [3].

The proofs will be published in the SIAM Journal on Control.

VI. EXAMPLE

As an application of the results in Section IV, we derive a L^p -stability ($1 \leq p \leq \infty$) criterion for the damped Mathieu equation with a forcing function. The stability regions in the parameter-plane are then compared with those obtained by Michael [17] and Parks [18] for the free damped Mathieu equation. The result is stated as Theorem 2.

Theorem 2.

Consider the forced damped Mathieu equation defined by

$$\ddot{y} + a\dot{y} + (b + \varepsilon \cos nt)y = u(t) \quad (22)$$

where a , b , ε , n are some finite constants with $a > 0$, $b > 0$ and $n > 0$.

Let k be a real number such that $b + k > 0$. Let $\lambda = \max\{|\varepsilon - k|, |\varepsilon + k|\}$.

If either

$$(i) \quad a^2 > 4(b + k) \text{ and } \lambda < \{(b + k)[a^2 - 4(b + k)]^{1/2}\}/a,$$

or

$$(ii) \quad a^2 < 4(b + k) \text{ and } \lambda < \{a[4(b + k) - a^2]^{1/2}\}/4,$$

then $u \in L^p$ ($1 \leq p \leq \infty$) implies $y \in L^p$. Furthermore, if $u \in L^2$, then $y \in L^2 \cap L^\infty$ and $y(t) \rightarrow 0$ as $t \rightarrow \infty$.

Comments.

1. In Theorem 2, if the linear time-varying gain $\psi(t) = \varepsilon \cos nt$ is replaced by a nonlinear time-varying gain $\varphi(\sigma, t)$ subjected to $|\varphi(\sigma, t)| \leq |\varepsilon| |\sigma|$, then the results of Theorem 2 still hold.
2. Fig. 2 shows various stability regions of Eq. (22) in the $b - \varepsilon$ plane with $a = 1$ obtained by Theorem 2 and by Parks [18]. With $k = 0$, the L^P -stability regions (i.e., the regions enclosed by the solid line) are $|\varepsilon| < [b(a^2 - 4b)^{1/2}]/a$ when $a^2 > 4b$ and $|\varepsilon| < [a(4b - a^2)^{1/2}]/4$ when $a^2 < 4b$. With $k = -b + 5a^2/16$, the L^P -stability region, (i.e., the region enclosed by the dash-dot line) is $|\varepsilon| < [b - (3a^2)/16]$. Note that the region obtained with $k = -b + 5a^2/16$ is much larger than those obtained with $k = 0$. This justifies the Comment 2 in Section IV that an appropriate choice of k will give a better stability result. The stability regions (i.e., the regions enclosed by dashed lines) obtained by Parks [18] for the free damped Mathieu equation (i.e., $u(t) \equiv 0$ in Eq. (22)) are $\varepsilon^2 < [a^2(4b - a^2)]/4$ when $a^2 < 4b$ and $\varepsilon^2 < b^2$ when $a^2 > 2b$. Note that even though there is a forcing function, the stability region obtained by Theorem 2 with $k = -b + 5a^2/16$ is not contained in that obtained by Parks for the zero-input case.

References.

- [1] Sandberg, I. W., "On the L^2 -boundedness of Solutions of Nonlinear Functional Equations", BSTJ, Vol. 43, July, 1964, pp. 1581-1599.
- [2] Zames, G., "On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems", Pt. I and II, IEEE Trans., Vol. AC-11, pp. 228-238, April, 1966, and pp. 465-476, July, 1966.
- [3] Sandberg, I. W., "A Condition for L^∞ -stability of Feedback Systems Containing a Single Time-Varying Nonlinear Element", BSTJ, July, 1964, pp. 1815-1817.
- [4] Zames, G., "Nonlinear Time-Varying Feedback Systems -- Conditions For L^∞ -boundedness Derived Using Conic Operators on Exponentially Weighted Spaces", Proc. 1965 Allerton Conference, pp. 460-471.
- [5] Bergen, A. R., R. P. Iwens and A. J. Rault, "On the Input-Output Stability of Nonlinear Feedback Systems", IEEE Trans., Vol. 1, AC-11, No. 4, Oct., 1966, pp. 742-745.
- [6] Desoer, C. A. and M. Y. Wu, "Stability of Linear Time-Invariant Systems", IEEE Trans., Vol. CT-15, No. 3, Sept., 1968.
- [7] Desoer, C. A. and M. Y. Wu, "Stability of Multiple-loop Feedback Linear Time-Invariant Systems", Journal of Mathematical Analysis and Applications, Vol. 22, Issue 3, June, 1968.

- [8] Desoer, C. A. and M. Y. Wu, "Stability of Linear Time-Invariant Systems", Proc. 2nd Princeton Conference on Information Sciences and Systems, pp. 12-14, March, 1968.
- [9] Chen, C. T., " L^p -stability of Linear Time-Varying Feedback Systems", J. SIAM Control, Vol. 6, No. 2, May, 1968, pp. 186-193.
- [10] Hille, E. and R. S. Phillips, Functional Analysis and Semi-Groups, (Revised edition) AMS, Providence, R.I., 1957.
- [11] Zames, G. and P. L. Falb, "Stability Conditions for Systems with Monotone and Slope Restricted Nonlinearities", SIAM J. on Control, Vol. 6, No. 1, pp. 89-108, Feb., 1968.
- [12] Dieudonné, J., Foundations of Modern Analysis, Academic Press, New York, 1960.
- [13] Sandberg, I. W., "Some Results on the Theory of Physical Systems Governed by Nonlinear Functional Equations", BSTJ, Vol. 44, pp. 871-898, May - June, 1965.
- [14] Desoer, C. A. Lecture notes for EECS 290D, Winter, 1968, University of California, Berkeley, California.
- [15] Schwartz, L., "Théorie des Distributions", (Revised edition) Hermann et Co., Paris, 1966.
- [16] Widder, D. V., "The Laplace Transform", Princeton University Press, 1946.
- [17] Michael, G. J., "Explicit Stability Criteria for the Damped Mathieu Equation", IEEE Trans. (correspondence), Vol. AC-12, No. 3, June, 1967, pp. 337-338.
- [18] Parks, P. C. and G. J. Michael, "Comment on Explicit Stability Criteria for the Damped Mathieu Equation", IEEE Trans. (correspondence), Vol. AC-13, No. 1, Feb., 1968, p. 129.

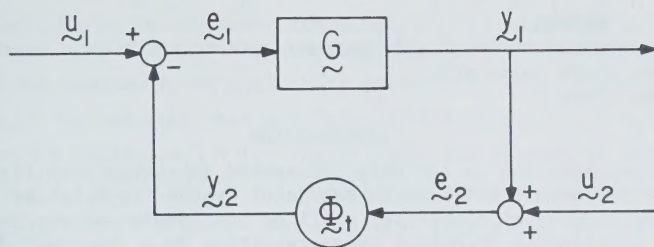


Fig. 1.

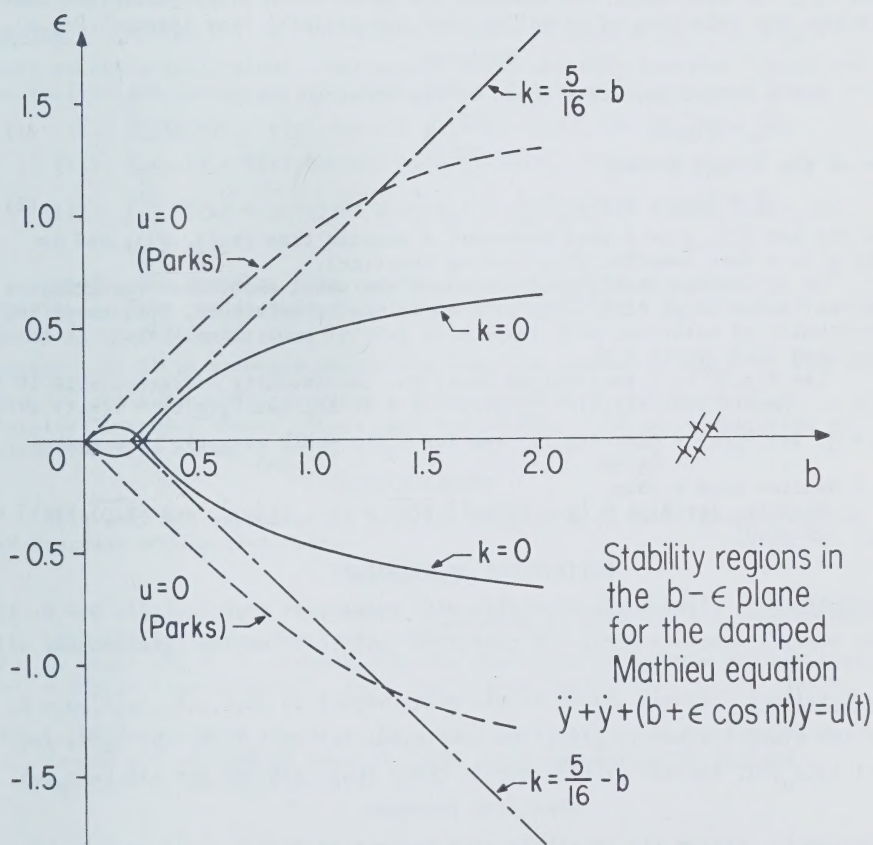


Fig. 2.

A. N. MICHEL
 Department of Electrical Engineering
 Iowa State University
 Ames, Iowa

INTRODUCTION

In practice one is not only interested in system stability (e.g. in the sense of Lyapunov), but also in bounds of system trajectories. A system could for example be stable and still be completely useless because it may exhibit undesirable transient characteristics (e.g. too much overshoot). Thus it may be useful to consider the stability of systems with respect to certain subsets of the state space which are selected a priori in a given problem. Problems along these lines have been suggested in [1] and have been considered, among others, for the case of finite time stability in [2] and [3]. In this paper two theorems are given which yield sufficient conditions for this type of stability over the infinite time interval $[t_0, \infty)$.

NOTATION

Let X denote the state space of the unforced system

$$\dot{\underline{x}} = \underline{f}(\underline{x}, t) \quad (1)$$

or of the forced system

$$\dot{\underline{x}} = \underline{f}(\underline{x}, t) + \underline{u}(\underline{x}, t). \quad (2)$$

In (1) and (2), $\underline{x}$ is a real n -vector, t denotes time ($t \in [t_0, \infty)$), and in (2) $\underline{u}$ is a real n -vector (the forcing function).

It is assumed that $\underline{f}(\underline{x}, t)$ possesses the usual smoothness requirements so that there is no difficulty with questions of existence, uniqueness and continuity of solutions with respect to initial conditions. Also, it is not required that $\underline{f}(\underline{0}, t) = \underline{0}$.

Let $V(\underline{x}, t)$ be a real-valued function, continuously differentiable in $\underline{x}$ and t . Recall that $\dot{V}(\underline{x}, t) = \nabla V(\underline{x}, t) \cdot \dot{\underline{x}} + \partial V / \partial t$, let $\dot{V}_f(\underline{x}, t) = \dot{V}(\underline{x}, t)$ with $\underline{u} = \underline{0}$, let $V_M^\alpha(t) = \max_{\|\underline{x}\|=\alpha} V(\underline{x}, t)$, and let $V_m^\beta(t) = \min_{\|\underline{x}\|=\beta} V(\underline{x}, t)$. The symbol $\|\cdot\|$ denotes here a norm.

Finally, let $B(\alpha) = \{\underline{x}: \|\underline{x}\| < \alpha\}$, $\overline{B(\alpha)} = \{\underline{x}: \|\underline{x}\| \leq \alpha\}$, and $[\overline{B(\beta)} - B(\alpha)] = \{\underline{x}: \alpha \leq \|\underline{x}\| \leq \beta\}$.

DEFINITIONS OF STABILITY

Definition 1: System (1) is stable with respect to $(\alpha, \beta, t_0, \|\cdot\|)$, $\alpha \leq \beta$, if for every trajectory $\underline{x}(t)$, the condition $\|\underline{x}(t_0)\| < \alpha$ implies $\|\underline{x}(t)\| < \beta$ for all $t \in [t_0, \infty)$.

Definition 2: System (2) is stable with respect to $(\alpha, \beta, \epsilon, t_0, \|\cdot\|)$, $\alpha \leq \beta$, if for every trajectory $\underline{x}(t)$, the conditions (a) $\|\underline{x}(t_0)\| < \alpha$, (b) $\|\underline{u}\| \leq \epsilon$ for all $t \in [t_0, \infty)$, for all $\underline{x} \in [\overline{B(\beta)} - B(\alpha)]$, imply that $\|\underline{x}(t)\| < \beta$ for all $t \in [t_0, \infty)$.

STABILITY THEOREMS

Theorem 1: System (1) is stable with respect to $(\alpha, \beta, t_0, \|\cdot\|)$, $\alpha < \beta$, if there exists a real-valued continuously differentiable function $V(\underline{x}, t)$ and a real-valued function $\psi(t)$ integrable over $[t_0, \infty)$, such that

- (i) $\dot{V}(\underline{x}, t) < \psi(t)$ for all $\underline{x} \in [\overline{B(\beta)} - B(\alpha)]$, for all $t \in [t_0, \infty)$

(ii) $\int_{t_1}^{t_2} \psi(t) dt \leq V_m^\beta(t_2) - V_M^\alpha(t_1)$, for all $t_1, t_2 \in [t_0, \infty)$, $t_2 > t_1$

Proof: Let $\underline{x}(t)$ be an arbitrary trajectory of (1) such that $\|\underline{x}(t_0)\| < \alpha$. Assume that there exists a $t_2 \in [t_0, \infty)$, the first point such that $\|\underline{x}(t_2)\| = \beta$. By the continuity of $\underline{x}(t)$ there exists a $t_1 < t_2$, $t_1 \in [t_0, \infty)$, such that $\|\underline{x}(t_1)\| = \alpha$ and such that $\alpha < \|\underline{x}(t)\| < \beta$ for all $t_1 < t < t_2$. Now write $V[\underline{x}(t_2), t_2] = V[\underline{x}(t_1), t_1] + \int_{t_1}^{t_2} \dot{V}[\underline{x}(t), t] dt$, and in view of (i) and $V_M^\alpha(t)$, obtain $V[\underline{x}(t_2), t_2] < V_M^\alpha(t_1) + \int_{t_1}^{t_2} \psi(t) dt$. Moreover, in view of (ii), observe that $V[\underline{x}(t_2), t_2] < V_m^\beta(t_2)$. But this last inequality implies that $\|\underline{x}(t_2)\| \neq \beta$, a contradiction to the original assumption. Hence, there does not exist a t_2 as indicated above and $\|\underline{x}(t)\| < \beta$ for all $t \in [t_0, \infty)$. Since this argument is independent of the exact value of $\underline{x}(t_0)$ and the particular trajectory chosen, the theorem is proved.

Theorem 2: System (2) is stable with respect to $(\alpha, \beta, \epsilon, t_0, \|\cdot\|)$, $\alpha < \beta$, if there exists a real-valued continuously differentiable function $V(\underline{x}, t)$ and two real-valued functions $\psi(t)$, $\xi(t)$ integrable over $[t_0, \infty)$, such that

- (i) $\|\nabla V(\underline{x}, t)\| \leq \psi(t)$ for all $\underline{x} \in [\overline{B(\beta)} - B(\alpha)]$, for all $t \in [t_0, \infty)$
- (ii) $\dot{V}_f(\underline{x}, t) < \xi(t)$ for all $\underline{x} \in [\overline{B(\beta)} - B(\alpha)]$, for all $t \in [t_0, \infty)$
- (iii) $\int_{t_1}^{t_2} [\xi(t) + \epsilon \psi(t)] dt \leq V_m^\beta(t_2) - V_M^\alpha(t_1)$, for all $t_1, t_2 \in [t_0, \infty)$, $t_2 > t_1$

Proof: The proof of this theorem is identical to the proof of Theorem 1. Here use is made of the identity $\dot{V}[\underline{x}(t), t] = \dot{V}_f[\underline{x}(t), t] + \nabla V[\underline{x}(t), t] \cdot \underline{u}[\underline{x}(t), t]$.

Remarks: 1. It is strongly emphasized that the numbers α , β , ϵ , t_0 and the norm $\|\cdot\|$ are selected a priori in a given problem, 2. Contrary to the classical Lyapunov theory, there does not appear to be any requirement of definiteness on either V or $\dot{V}$.

ACKNOWLEDGEMENT

This work was supported in part by the office of Naval Research under ONR Contract N00014-68-A-0162.

REFERENCES

- [1] J. P. LaSalle and S. Lefschetz, Stability by Liapunov's Direct Method with Applications, New York: Akademik, 1961.
- [2] L. Weiss and E. F. Infante, "On the Stability of Systems Defined Over a Finite Time Interval", Proc. Nat'l Acad. Sci., Vol. 54, pp. 44-48, July, 1965.
- [3] L. Weiss and E. F. Infante, "Finite Time Stability Under Perturbing Forces and on Product Spaces", IEEE Trans. on Automatic Control, Vol. Ac-12, No. 1, February, 1967.

K.W. HEIZER

Southern Methodist University & Collins Radio

Dallas, Texas

ABSTRACT

A set of transfer functions that can be realized with a given RC distributed circuit is used to form a set of real functions including both the real and the imaginary parts. The Gram-Schmitt process is used to form a set of orthogonal functions from the real functions. The desired transfer function is then approximated using the orthogonal functions. Thus, the corresponding coefficients of the original functions are determined and the design of the distributed circuit is complete. The method is fast since it obtains the best fit in one pass. However, if the desired function is not RC in nature, results can be obtained that are not practical.

INTRODUCTION

The general problem of synthesis of a given response in both the time and frequency domain by controlling the value of the resistance and capacity as a function of distance has been treated and an iterative solution given. [1] The problem presented here is one in which the overall structural boundary is fixed and the transfer function is determined by the cut of a conducting electrode within the boundary. Given a voltage transfer function, the shape of the cut can be machine plotted. Thus a mask for thin film construction is formed providing a completely automated design.

STRUCTURE

The general form of the structure is shown in Figure 1. This structure has been analyzed (2) and the short-circuit parameters of interest here are:

$$Y_{11} = Y_{22} = \frac{W \gamma \coth \gamma L}{R} \quad (1)$$

$$-Y_{21} = W\gamma / R \sinh \gamma L \quad (2)$$

$$-Y_{23} = \frac{sC}{\sinh \gamma L} \int_0^L f(x) \sinh \gamma (L-x) dx \quad (3)$$

where $\gamma = \sqrt{RCs}$

R = resistance on ohms per square

C = capacity from the resistive sheet to the electrodes in

farads per unit area

L = length and W = width

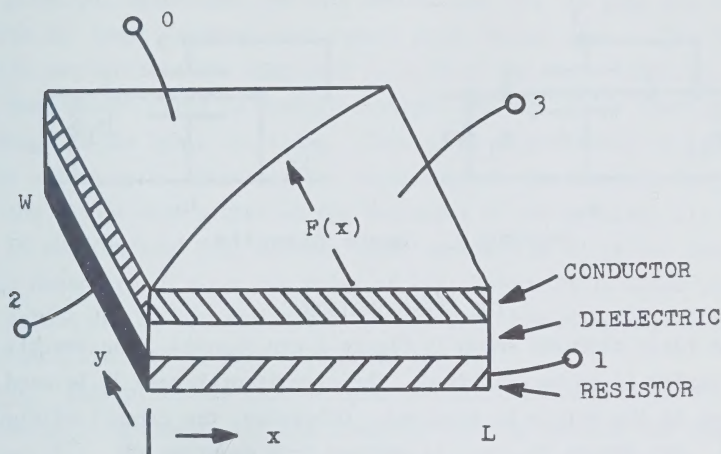


FIGURE 1. Basic Structure

It should be noted that if $f(x)$ is given as a linear combination of simple functions of x , $-y_{23}$ can be given as the same linear combination of the corresponding short-circuit transfer admittances. One very useful form is

$$f(x) = Wx/L + W \sum_{M=1}^{N-1} a_m \sin \frac{m\pi x}{L} \quad (4)$$

for which

$$-y_{21} - y_{23} = \frac{W}{RL} \left[1 + \pi \sum_m \frac{m a_m s}{s + \lambda_n} \right] \quad (5)$$

$$\text{where } \lambda_n = \frac{n^2 \pi^2}{L^2 RC} \quad (6)$$

Another useful form is a sequence of delayed pulses.

$$f(x) = \sum_m a_m \left[U(x - x_m) - U(x - x_{m+1}) \right] \quad (7)$$

where $U(x)$ is the unit step.

The corresponding transfer admittance is

$$-y_{23} = \frac{\sum_m a_m sC \left[\cosh \gamma (L - x_m) - \cosh \gamma (L - x_{m+1}) \right]}{\sinh \gamma L} \quad (8)$$

The transfer function in this case is not an easy one to evaluate, but it is an easy one to constrain. Other functions of x are also useful for some problems.

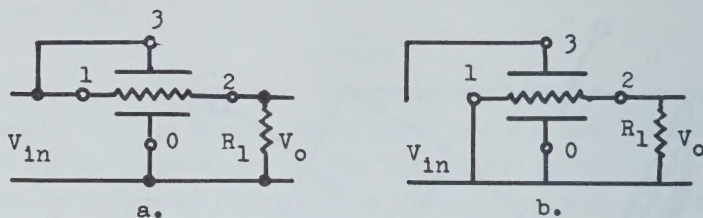


FIGURE 2. Basic Circuits

THE APPROXIMATION

The basic circuits shown in Figure 2 are assumed. The overall voltage transfer is to be specified. The circuit of Figure 2b is used if a zero at the origin is required. Otherwise, the circuit of Figure 2a is used. For Figure 2b, $-y_{21}$ is omitted from equation (9).

$$\frac{V_o}{V_{in}} = \frac{-y_{21} - y_{23}}{y_{22} + 1/R_1} \quad (9)$$

Once the basic structure is chosen, the value of y_{11} is known and the only unknown in equation (9) is $-y_{23}$. The approximation problem is reduced to that of approximating the desired y_{23} .

As stated, the short-circuit transfer admittance y_{23} can be represented as a linear combination of basic transfer admittances. It is thus possible to form orthogonal functions from the basic transfer admittances. The desired function can be expressed as a linear combination of the orthogonal functions and then as a linear combination of the basic functions. The method is to evaluate the basic functions at M discrete frequencies. These discrete values of frequency do not have to be uniformly spaced, but once chosen, they remain fixed for the remainder of the approximation. To each of the discrete frequencies a weight may be assigned. This weight is used in the inner-product and thus in the determination of the mean-square error.

The basic functions and the desired function are complex. Each is put in real form as follows. The function is represented as $R + jX$ at f_1 . A real function is generated as $F(f_1) = R + X$ and $F(-f_1) = R - X$. Thus the complex functions are replaced by real functions that are defined at both positive and negative frequencies. The computer storage space

is the same for each form. The orthogonalization and the approximation are carried out with these real functions. Thus, both the real and the imaginary parts of the desired transfer function are approximated.

The principal advantages for this method are: (1) It gets the response with the least weighted mean square error in one pass. (2) Once the initial computations are completed using N of the basic functions, the weighted mean square error is readily obtained for any linear combination of N or fewer of the basic functions. This saves a great deal of time when it is necessary to constrain the excursions of the function $f(x)$. Although the bounds were picked at the beginning of the problem, all is not lost if the function $f(x)$ exceeds these bounds. If it is too large, it is only necessary to alter the value of $-y_{21}$ before it is added in. Negative values can be accommodated by means of an inverting amplifier (3) or by two such circuits to give a balanced input.

Limitations. This procedure depends upon the basic admittance functions being linearly independent. If the frequency band is outside the range of the available poles, or if too many of the functions are defined, some may be almost linearly dependent on others. This in turn causes the amplitude of one or more of the orthogonal functions to have a very small amplitude and may require a large coefficient in the approximation.

EXAMPLE

Since the approximation is only concerned with the value of y_{23} , this could serve as the starting point of the problem. However, in order that the errors be compared to the overall short-circuit transfer admittance, the approximation is for $-y_{21} - y_{23}$ with equation (5) defining the basic functions. Let the desired transfer admittance be

$$-y_{21} - y_{23} = \frac{s^2 + 4s + 8}{(s + 1)(s + 2)(s + 3)} \quad (10)$$

to be approximated over a 12 octave range with 2 points per octave beginning at $W = 0.1$.

The poles of this function are such that an exact synthesis requires a separate unit for each pole. However, a higher order function which has poles that can be obtained from one unit is used with only a very small error.

The results for $\lambda_1 = 0.1$ are given in Table I for different values of N . N is the number of basic functions used. It should be observed from equation (4) that the number of sine terms is $N - 1$.

TABLE I

	N = 10	N = 9	N = 8	N = 7	N = 6
m.s.e.	.000000	.000000	.000001	.000002	.000032
a_1	-.001183	-.0015524	-.0022311	-.0039117	-.011808
a_2	.005888	.0071096	.0092159	.013823	.033165
a_3	-.15131	-.15675	-.16497	-.17995	-.23005
a_4	-.044408	-.025321	-.001565	.032096	.11253
a_5	.17765	.13091	.085849	.040341	-.027035
a_6	-.18265	-.10713	-.055061	-.022235	
a_7	.11829	.042491	.0096150		
a_8	-.051102	-.0086318			
a_9	.010091				

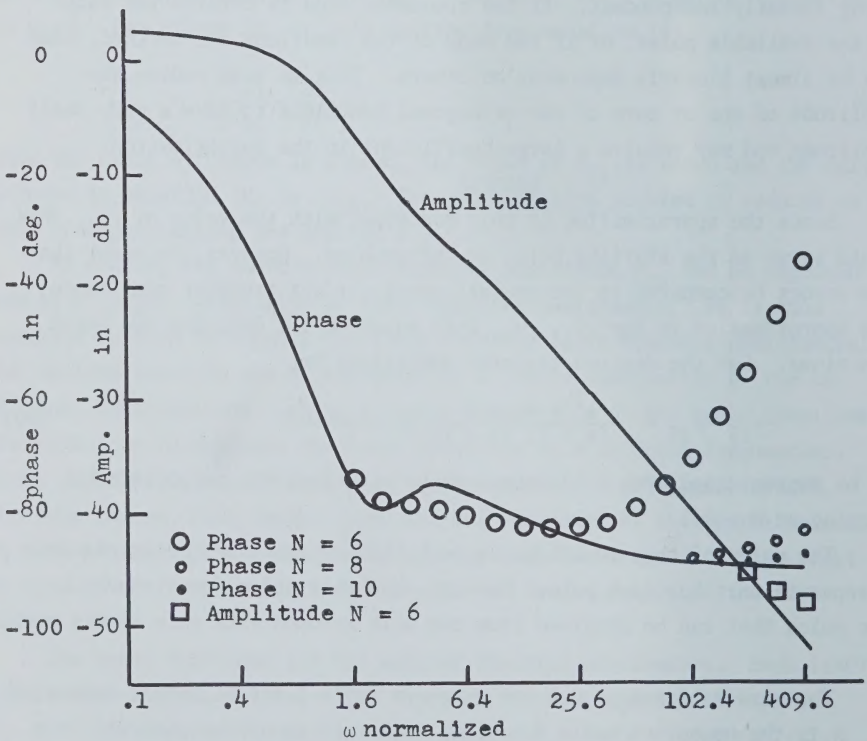


FIGURE 3. The curve is the desired value. Only the points that are significantly different are marked.

The mean square error (m.s.e.) shown in Table I is not zero for $N = 10$ and $N = 9$ but is less than the value for $n = 8$. The average value of the function approximated was 0.301328 with a maximum value of 1.2288. The weight function was taken as unity for all points.

It should be remembered that the approximation was carried out on the real and imaginary parts of the function. The results are shown in Figure 3. The magnitude is plotted in db. When the function is small the errors shown up, particularly in the phase.

The very nature of the computations as outlined in the Appendix give rise to considerable roundoff errors. It has been found necessary to use computer word lengths greater than 36 bits if the approximation is to hold for more than a 30 db range. This problem was solved on a Univac 1108 computer using double precision (2 x 36 bit words). The time required for $N = 10, 8, 6, 4, 2$ was 7 seconds.

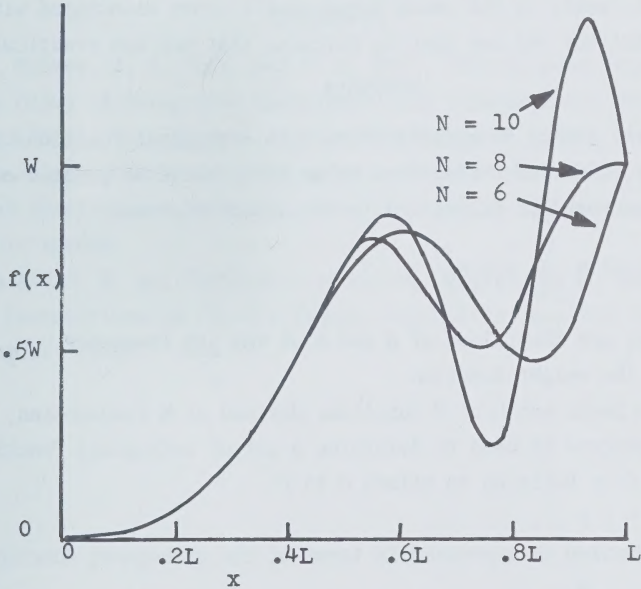


FIGURE 4. The cut as a function of x .

Figure 4 shows the shape of the cut for three different values of N . It may be observed that for $N = 10$, $f(x)$ exceeds the width W . This is generally the case when too much is demanded. In particular, $f(x)$ is likely to exceed both the upper and lower bounds when the transfer function is not of RC nature. The same is true if too large a value for N is used. For this reason it is usually better to first solve the problem for a

reasonably large value of N (10-15). Solutions may then be obtained from this for smaller values of N by reducing the Fourier coefficients of the higher terms in the orthogonal representation to zero. In this problem $N = 8$ is the best approximation that remains in bounds. There is no advantage in using N less than 8 for this problem.

It may be deduced correctly from Figure 4 that the circuit is more sensitive to the shape of $f(x)$ for small values of x .

CONCLUSIONS

A method using orthogonal functions has been described which can be used to specify an RC distributed circuit to approximate a given voltage transfer function, or a short-circuit transfer admittance. The approximation is on both the amplitude and phase. The method usually gives good results for RC type functions. However, if a close fit is required, some functions will result in the usual large coefficients associated with orthogonal functions and may specify circuits that are not practical.

APPENDIX

The general method of approximation with orthogonal functions is well known; however, a few of the options taken here should be pointed out. First, the inner-product is defined in the discrete sense.

$$(d, h) = \sum_{j=1}^M d_j h_j w_j \quad (1A)$$

where d_j and h_j are the values of d and h at the j th frequency. w_j is the component of the weight function.

Given the basic set G of N functions defined at M frequencies, the Gram-Schmitt process is used to determine a set of orthogonal functions P . An $N \times N$ matrix C is built up to relate G to P

$$G = C P \quad (2A)$$

The desired function is expressed in terms of the orthogonal functions as

$$F_j = \sum_{i=1}^N a_i p_j^i \quad (3A)$$

The a_i 's are the Fourier coefficients and p_j^i is the i th orthogonal function evaluated at the j th frequency. This function is then expressed by means of equation (2A) as

$$F_j = \sum_{i=1}^N b_i g_j^i \quad (4A)$$

The coefficients of the desired function $f(x)$ are the coefficients b_i .

If it becomes necessary to alter the coefficients in order to stay in bounds such that

$$F_j = \sum_{i=1}^N d_i p_j^i, \quad (5A)$$

the new mean square error can be determined immediately by the relation

$$\text{m.s.e.} = \frac{1}{M} \sum_{i=1}^N (a_i - d_i)^2 A_i + E. \quad (6A)$$

E is the fixed error due to the failure of the original functional space to completely span the desired function. It is the original error. A_i is the squared amplitude of the i th orthogonal function.

REFERENCES

- 1 R. A. Rohrer, J. A. Resh, and R. A. Hoyt, "Distributed Network Synthesis for a Class of Integrated Circuits," IEEE International Convention Record (USA), Vol. 13, pt. 7, pp. 119-126; March, 1965.
- 2 K. W. Heizer, "Analysis of Rectangular RC Distributed Circuits with Shaped Electrodes," To be published in The IEEE Transactions on Circuit Theory.
- 3 Y. Fu and J. S. Fu, "Synthesis of Active Distributed RC Networks," IEEE Transactions on Circuit Theory, Vol. CT-13, pp. 259-264; September, 1966.

TRANSIENT RESPONSE OF A CLASS OF ACTIVE LUMPED-DISTRIBUTED SYSTEMS

J. J. BOURQUIN and J. S. TZENG
University of Missouri -- Rolla
Rolla, Missouri

ABSTRACT

The transient response is investigated for that class of lumped-distributed systems for which transfer functions are rational in $w = \exp(\sqrt{s}Ts)$. By comparing the resulting plots with similar plots for transfer functions rational in $\theta = \sqrt{s}$, considerable insight is provided into the transient characteristics of an important class of transfer functions meromorphic in θ .

INTRODUCTION

Impulse and step responses are investigated for that class of active lumped-distributed systems for which the transfer function is rational in $w = \exp(\sqrt{s}Ts)$, and the degree of the numerator is no greater than that of the denominator; T is a real positive constant, and s is the complex frequency variable. Physical examples of this class include the uniform distributed RC network (UDRC) with open-circuit, short-circuit, or matched terminations, the distributed-lag element (DL), and the constant-gain amplifier, open- and closed-loop combinations of which yield a variety of possible configurations.

Transfer functions of this class are generally double valued in s , while that subclass of single-valued functions is a special case. As is shown in the next section, it is incorrect to analyze the general case by poles and zeros in the s plane; only when the transfer function is single valued do the infinite number of poles and residues have the usual interpretation. Because of this, in addition to the simplicity of the pole-zero configuration for this class in the w plane, the responses are computer plotted for various real and complex-conjugate w -plane pole locations.

Consider that broader class of systems for which the transfer function is of the form

$$T(s) = \frac{A_1 G_1(\theta) H_1(w) + A_2 G_2(\theta) H_2(w)}{A_3 G_3(\theta) H_3(w) + A_4 G_4(\theta) H_4(w)} \quad (1)$$

where $\theta = \sqrt{s}$, $G_k(\theta)$ and $H_k(w)$ are polynomials of degree m_k and n_k , respectively, and $m_4 \geq \sup\{m_1, m_2, m_3\}$ while $n_4 \geq \sup\{n_1, n_2, n_3\}$. Physical examples of this class include those listed above with the addition of the UDRC terminated in any impedance function rational in s or θ , elements with transfer functions rational in s or θ , frequency dependent amplifiers, etc.

As in the previous class, the transfer function in Equation (1) is, in the general case, double valued in s . Although the special single-valued case can be analyzed in the s plane by, for instance, the root-locus method¹, the general case must be analyzed in the θ plane; $T(s)$ is a meromorphic function of θ and can be correctly analyzed only by its θ -plane poles and zeros. A root-locus method has been devised utilizing the θ and w planes by which all of the infinite number of θ -plane poles and zeros can be located; this method can also be applied to the single-valued case and is usually simpler than the s -plane root-locus plot².

But most important to this discussion is that the infinite number of large- θ poles and zeros of $T(s)$ approach, as θ becomes large, θ -plane points which are mappings of a finite number of w -plane points². Thus, considerable insight should be provided into this class of systems, as well as the

former, especially when considered in light of a similar set of responses plotted for transfer functions rational in Θ^3 , such as those shown in Figure 1. In the next section the impulse response for a simple pole in the w plane is derived. Frequency is normalized by letting $T = 1$.

IMPULSE RESPONSE

In applying the Laplace transformation inversion integral to

$$h(t) = L^{-1}[H(s)] = L^{-1}[(w-a)^{-1}] = L^{-1}[(e^{\sqrt{s}-a})^{-1}], \quad (2)$$

the impulse response of a simple w -plane pole, it is convenient to make the denominator single valued in s . For this reason, let

$$H(s) = \frac{1}{w-a} \cdot \frac{w^{-1}-a}{w^{-1}-a} = H_1(s) - H_2(s), \quad (3)$$

where

$$H_1(s) = \frac{-a^{-1}}{(w-a)(w-a^{-1})}, \quad (4)$$

and

$$H_2(s) = \frac{-w}{(w-a)(w-a^{-1})}; \quad (5)$$

$H_1(s)$ is double valued in s , but $H_2(s)$ is single valued.

The inversion of $H_1(s)$,

$$h_1(t) = \lim_{R \rightarrow \infty} \frac{1}{j2\pi} \int_{c+jR}^{c+jR} \frac{-a^{-1} e^{st} ds}{(w-a)(w-a^{-1})}, \quad (6)$$

is considered first. The poles of $H_1(s)$ are located in the w plane at points

$$a = re^{j\alpha} \quad \text{and} \quad a^{-1} = r^{-1} e^{-j\alpha}, \quad (7)$$

which map onto the s plane at points

$$s = (\ln_n a)^2$$

where

$$\ln_n a \triangleq \ln r + j(\alpha + 2n\pi); \quad n = 0, \pm 1, \pm 2, \dots \quad (8)$$

These s -plane poles occur along a parabola (Figure 2).

In evaluating the inversion integral in Equation (6), the branch point of $H_1(s)$ at the origin of the s plane must be taken into account; the branch cut is taken along the negative real axis, and the contour of integration is deformed to avoid intersection with the branch cut, as shown in Figure 2.

There exists an infinite sequence of increasing parabolic contours $C_1, C_2, \dots, C_k, \dots$, none of which intersect with any of the poles of $H_1(s)$, such that²

$$|H_1(s \text{ on } C_k)| \leq \delta_k,$$

and $\delta_k \rightarrow 0$ as $k \rightarrow \infty$.

$$\text{Therefore}^{4,5} \quad \lim_{k \rightarrow \infty} \int_{C_k} H_1(s) e^{st} ds = 0. \quad (9)$$

It can be shown that the integral along the infinitesimal circular contour around the branch point is zero. The integral evaluated along both sides of the branch cut is²

$$\int_{\text{branch cut}} H_1(s) e^{st} ds = -j\sqrt{\pi} t^{-\frac{3}{2}} \sum_{n=1}^{\infty} n a^{n-1} e^{-\frac{n^2}{4t}} \quad (10)$$

The residue for the n^{th} pole at $s = (\ln_n a)^2$ is

$$R_n = 2(\ln_n a)(a^{-1} - a)^{-1} e^{(\ln_n a)^2 t} \quad (11)$$

Therefore, by the theory of residues

$$h_1(t) = \frac{t^{-\frac{3}{2}}}{2\sqrt{\pi}} \sum_{n=1}^{\infty} n a^{n-1} e^{-\frac{n^2}{4t}} + \frac{2}{a^{-1} - a} \sum_{n=-\infty}^{\infty} (\ln_n a) e^{(\ln_n a)^2 t} \quad (12)$$

By a similar analysis $h_2(t)$ can be determined. However, since $H_2(s)$ is single valued in s , there is no component in $h_2(t)$ like that in Equation (10), and

$$h_2(t) = 2(a^{-1} - a)^{-1} \sum_{n=-\infty}^{\infty} (\ln_n a) e^{(\ln_n a)^2 t} \quad (13)$$

From Equation (3) the desired inverse transform is

$$h(t) = h_1(t) - h_2(t) = \frac{t^{-\frac{3}{2}}}{2\sqrt{\pi}} \sum_{n=1}^{\infty} n a^{n-1} e^{-\frac{n^2}{4t}} \quad (14)$$

One observes that $h_2(t)$, the inverse transform of a single-valued function of s , is an infinite series of terms, each of which is attributable to a simple pole in the s plane; but in $h(t)$, the impulse response of a simple w -plane pole and the inverse transform of a double-valued function of s , all such terms have subtracted out. Then the conclusion is that for the general case of a transfer function rational in w , poles plotted in the s plane and the corresponding residues are meaningless.

In the next section the step response of a simple pole in the w plane is discussed.

STEP RESPONSE

By a method similar to that described in the previous section, the step response of $H(s)$ in Equation (2) is

$$v(t) = L^{-1}[s^{-1}(w-a)^{-1}] = \sum_{n=1}^{\infty} a^{n-1} \operatorname{erfc}\left(\frac{n}{2} \sqrt{t^{-1}}\right) \quad (15)$$

As was true for the corresponding impulse response in Equation (14), none of the terms in Equation (15) is directly attributable to any of the s -plane poles of $H(s)$. Again, s -plane poles and residues are meaningless.

The computer plots of the impulse and step responses for simple w -plane poles are discussed in the next section.

PLOTS OF THE TRANSIENT RESPONSES

The transient responses described in Equations (14) and (15) were plotted⁶ on an IBM 360/50 digital computer for various real and complex-

conjugate w-plane pole locations both within and outside the region of bounded input-bounded output stability, i.e., that region within the curve $\exp(|\alpha| + j\alpha)$, where $-\pi \leq \alpha = \arg w \leq \pi$, shown as the shaded area in Figure 3². For many of the families of curves discussed in this section, the response for $a = 0$ (i.e., $H(s) = \exp(-\sqrt{s})$) has been included as a matter of reference, since this is a case familiar to many. In Figures 4 and 5 are shown the impulse responses from Equation (14) for a simple pole on the positive and negative axes of the w plane.

The imaginary part of $h(t)$ in Equation (14) is plotted in Figures 6 through 9 for various values of r and α in Equation (7). These plots represent the impulse response of a transfer function of the form

$$\frac{\text{Im}(a)}{(w - a)(w - a^*)}, \quad (16)$$

i.e., with complex-conjugate poles and a zero-degree numerator.

The real part of $h(t)$ is plotted in Figures 10 through 12. The resulting plots represent the impulse response of the transfer function

$$\frac{w - \text{Re}(a)}{(w - a)(w - a^*)}, \quad (17)$$

i.e., with complex-conjugate poles and a first-degree numerator.

In Figures 13 and 14 Equation (15) is plotted for real positive and negative values of a , but each transient plot is magnitude scaled by a factor $1 - a$, i.e., normalized such that all stable responses approach unity in steady state. This normalization is for convenience in comparing various transient characteristics like rise time, overshoot, settling time, etc.

The imaginary part of $v(t)$ in Equation (15) is plotted in Figures 15 through 18. These plots are normalized by the factor

$$[1 - 2 \text{Re}(a) + |a|^2][\text{Im}(a)]^{-1}$$

and represent the step response of the transfer function

$$\frac{(1 - a)(1 - a^*)}{(w - a)(w - a^*)}. \quad (18)$$

Where appropriate in each of the aforesaid Figures, the value of

$$r_s = \exp(\alpha) \quad (19)$$

is given; that is, the boundary of the region of bounded input-bounded output stability is located for each angle investigated. In the next section the concept of dominant poles is discussed.

0-PLANE DOMINANT POLES

The 0-plane mapping of a simple w-plane pole at point a consists of an infinite sequence of poles at points $\ln_n a$ along a vertical line, as specified in Equation (8). $H(s)$ in Equation (2) can be expanded² in terms of these poles by the Mittag-Leffler expansion theorem to give

$$H(s) = a^{-1} \sum_{n=-\infty}^{\infty} (\theta - \ln_n a)^{-1}. \quad (20)$$

A transient response of a simple w-plane pole can then be expressed as the sum of the corresponding transient response of each of the infinite sequence of simple 0-plane poles mapped from the given w-plane pole.

One observes in Figure 1 that as ϕ approaches 90° from either direction the step response reaches steady more rapidly. And because of the time scale factor, the (stable) larger magnitude poles reach steady state more rapidly than do the shorter magnitude poles on the same radial line. Since the large- θ poles of $T(s)$ in Equation (1) approach the mappings of a finite number of w-plane poles, the aforesaid characteristics strongly suggest that the concept of dominant poles in the θ plane can be gainfully applied to transfer functions of the form $T(s)$.

The step responses in Figure 1 compared with those of Figures 15

through 18 provide an opportunity for qualitatively checking this concept. For convenience, several w-plane poles are listed in the table below along with the dominant pole in the θ plane, the step response of which can be observed in Figure 1.

θ -PLANE DOMINANT POLES				
w-plane pole			θ -plane dominant pole	
α	r	Figure	ϕ	ρ^2
45°	1.94	17	50°	1.05
120°	5.81	18	50°	7.50
45°	1.58	17	60°	0.82
120°	3.36	18	60°	5.85
45°	1.33	17	70°	0.70
120°	2.14	18	70°	4.97
45°	1.00	17	90°	0.62
120°	1.00	18	90°	4.39
45°	0.26	17	150°	2.46
120°	0.03	18	150°	17.5

A comparison between the pairs of curves indicated above shows that the concept of θ -plane dominant poles can indeed be useful in the design of a system for which the transfer function is $T(s)$ in Equation (1).

CONCLUSION

The transient response of a simple w-plane pole has been exhibited and compared with that of a simple θ -plane pole. For the general double-valued case of the transfer function $T(s)$ in Equation (1), poles and residues in the s plane are meaningless, and the concept of s-plane dominant poles is not applicable. Such transfer functions must be analyzed in the θ plane, wherein the concept of dominant poles is valid and useful.

REFERENCES

1. M. S. Ghausi and J. J. Kelly, Introduction to Distributed-Parameter Networks with Applications to Integrated Circuits, Holt, Rinehart, and Winston, Inc., New York, 1968.
2. J. J. Bourquin, "Stability of a Class of Lumped-Distributed Systems", Report R-373, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois, January, 1968.
3. M. B. Kilomeitseva and A. V. Netushil, "Transients in Automatic Control Systems with Irrational Transfer Functions", Automation and Remote Control, Vol. 26, No. 2, February 1965; pp. 353-358.
4. G. Doetsch, Handbuch der Laplace Transformation, Band I, Theory der Laplace Transformation, Verlag Birkhauser Basel, 1950.
5. G. Doetsch, Guide to the Application of Laplace Transforms, D. Van Nostrand Company Ltd., 1961.
6. J. S. Tzeng, "Transient Response of a Class of Distributed Systems", M. S. Thesis, U.M.R., October 1968.

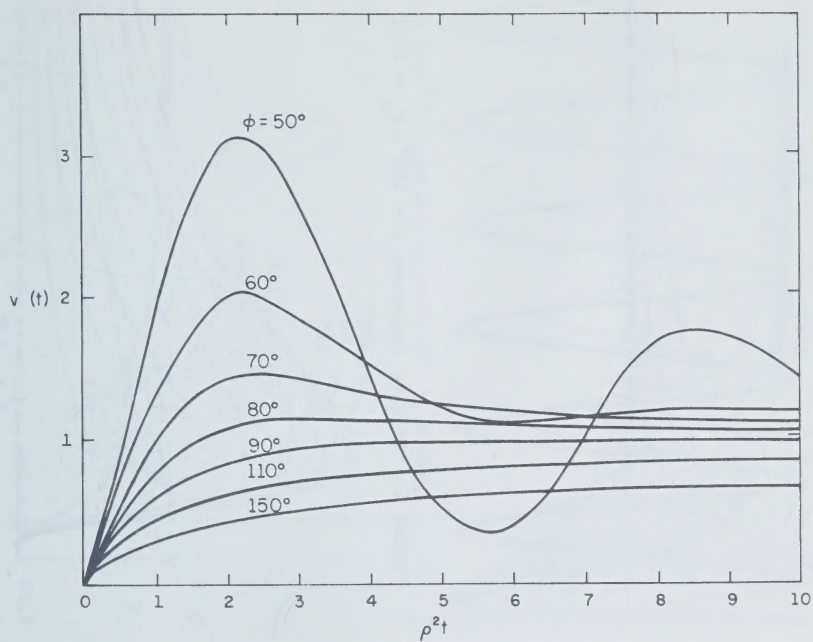


Figure 1. Normalized step response for $\frac{\rho^2}{(\theta-b)(\theta-b^*)}$; $\rho = |b|$;
 $\phi = \arg(b)$.

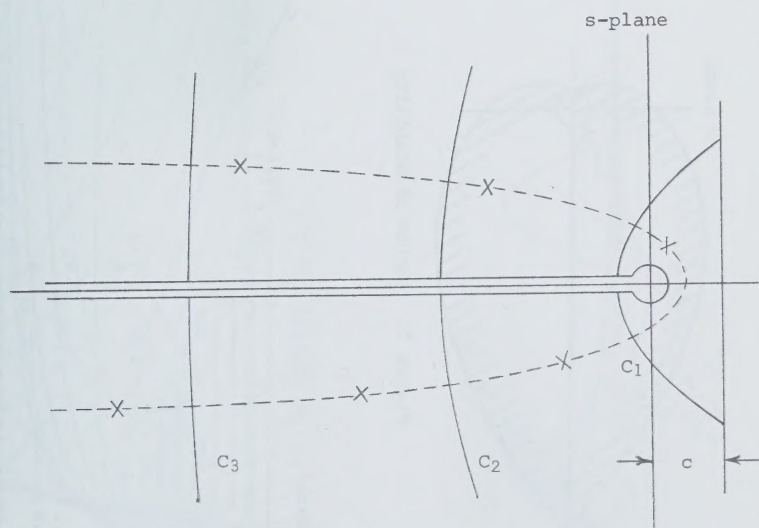


Figure 2. Contours of integration for finding $h_1(t)$.

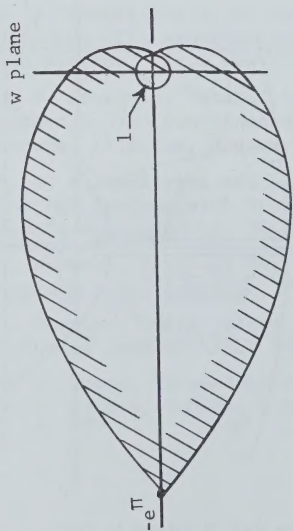


Figure 3. Region of stability

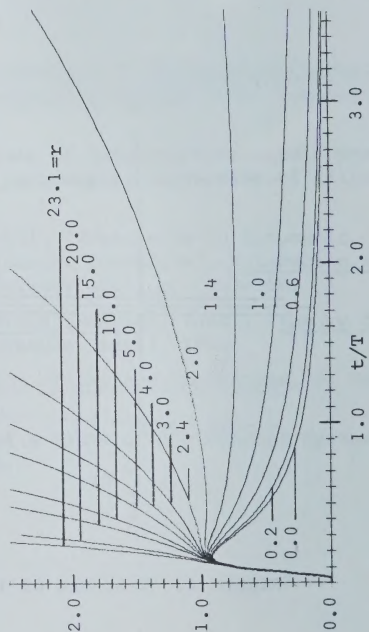


Figure 4. Impulse response; $\alpha = 0$; $r_s = 1.0$.

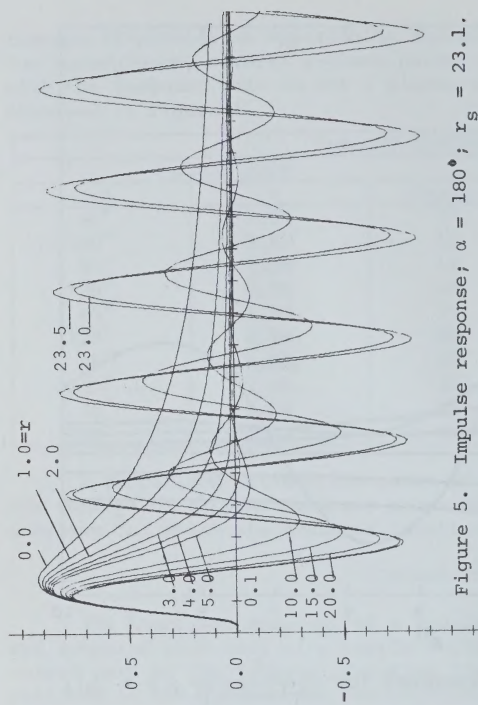


Figure 5. Impulse response; $\alpha = 180^\circ$; $r_s = 23.1$.

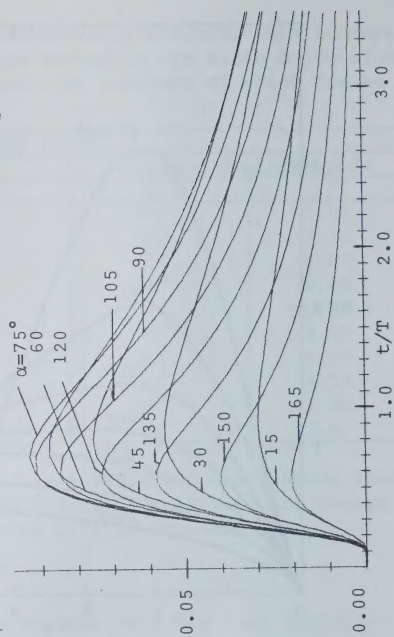


Figure 6. Im $[h(t)]$; $r = 0.4$.

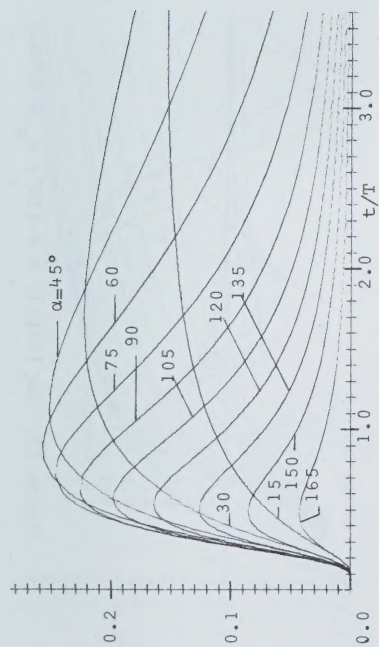


Figure 7. $\text{Im}[h(t)]$; $r = 1.0$.

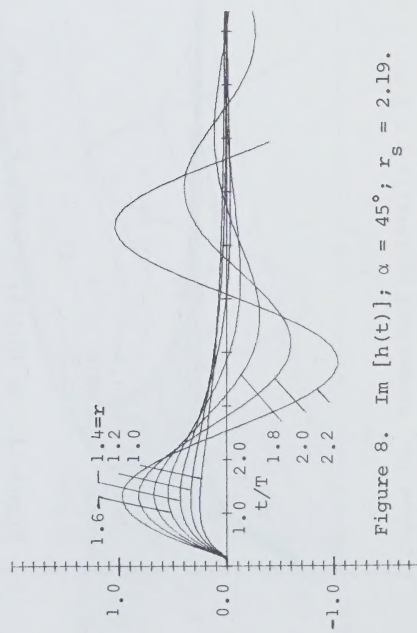


Figure 8. $\text{Im}[h(t)]$; $\alpha = 45^\circ$; $r_s = 2.19$.

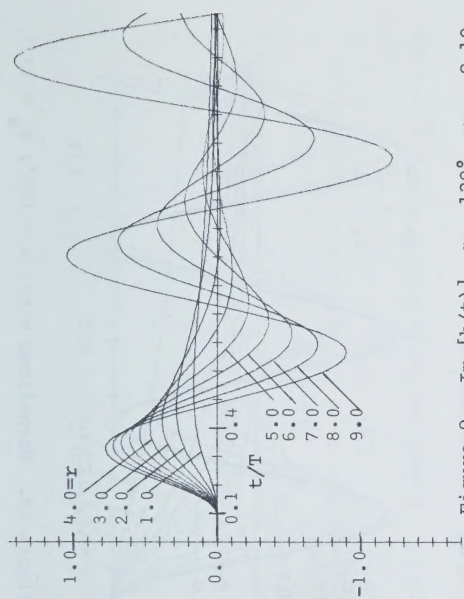


Figure 9. $\text{Im}[h(t)]$; $\alpha = 120^\circ$; $r_s = 8.10$

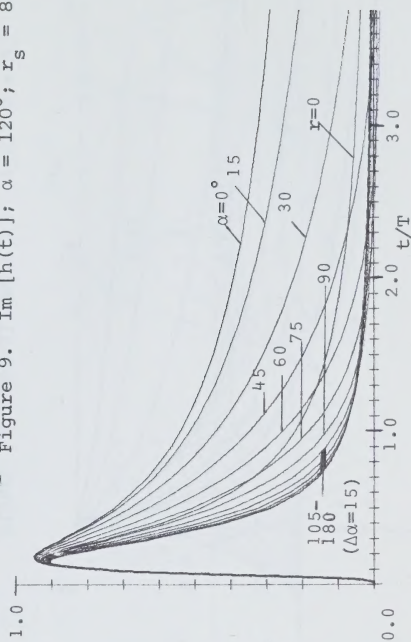


Figure 10. $\text{Re}[h(t)]$; $r = 1.0$.

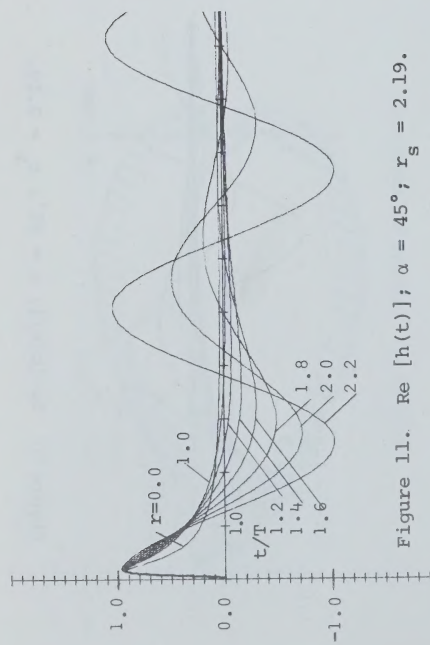


Figure 11. $\text{Re } [h(t)]$; $\alpha = 45^\circ$; $r_s = 2.19$.

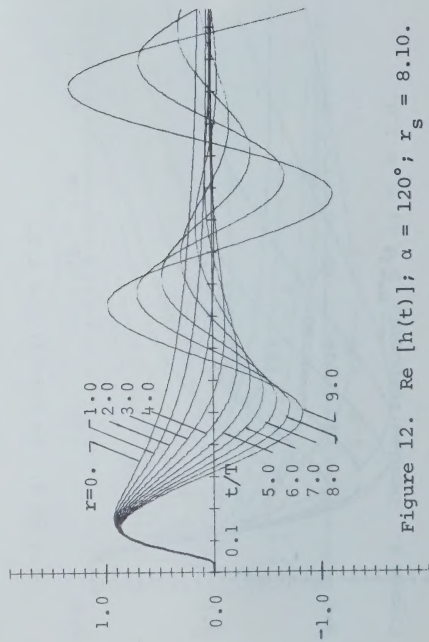


Figure 12. $\text{Re } [h(t)]$; $\alpha = 120^\circ$; $r_s = 8.10$.

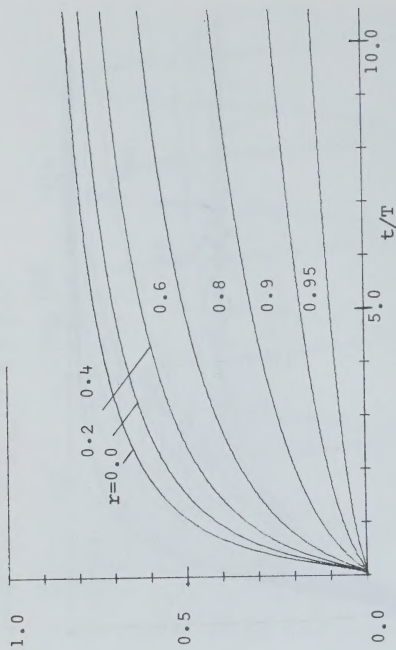


Figure 13. Normalized $v(t)$; $v(t)$; $\alpha = 0^\circ$.

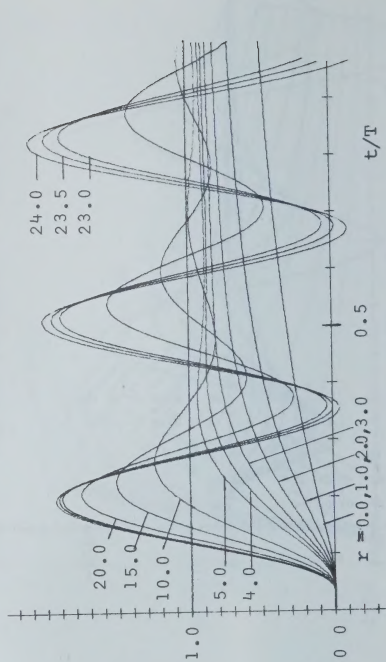


Figure 14. Normalized $v(t)$; $\alpha = 180^\circ$; $r_s = 23.1$.

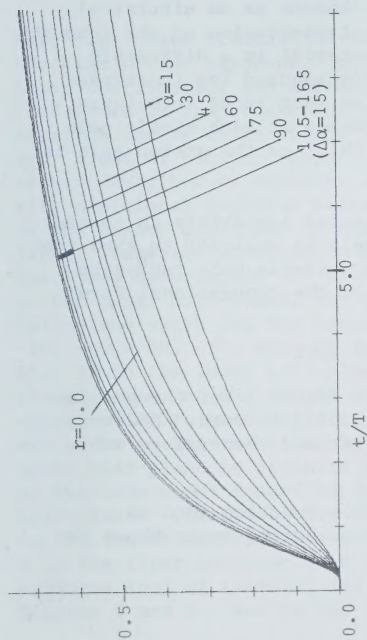


Figure 15. Normalized Im $[v(t)]$; $r = 0.4$.

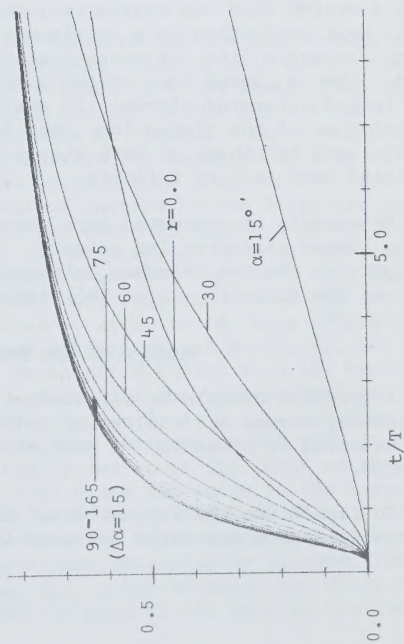


Figure 16. Normalized Im $[v(t)]$; $r = 1.0$.

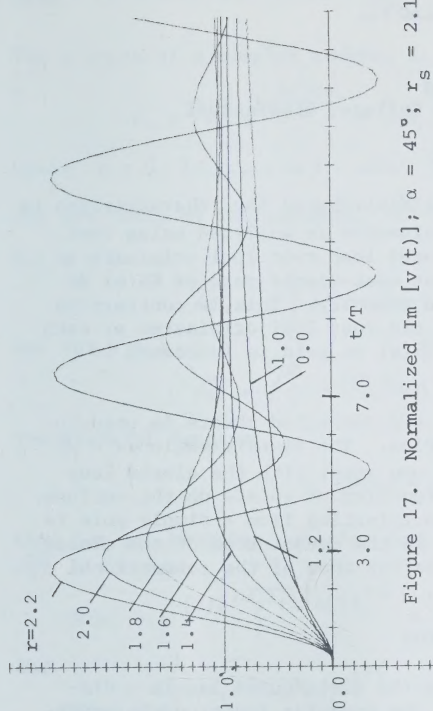


Figure 17. Normalized Im $[v(t)]$; $\alpha = 45^\circ$; $r_s = 2.19$.

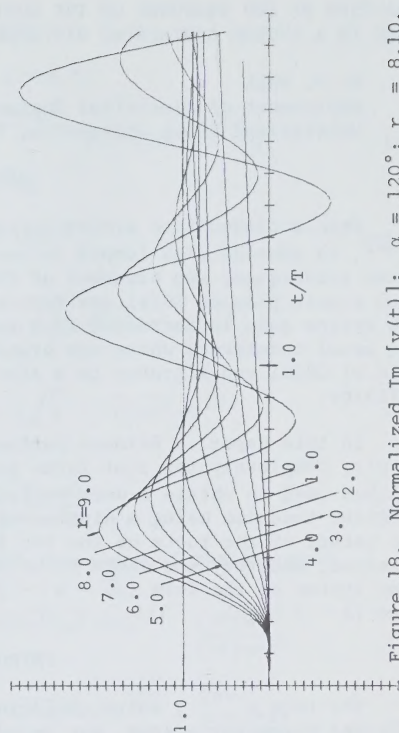


Figure 18. Normalized Im $[v(t)]$; $\alpha = 120^\circ$; $r_s = 8.10$.

ANALYSIS OF TWO BRANCHES OF THE ROOT LOCUS WHICH ORIGINATE FROM A SIMPLE POLE IN A SYSTEM CONTAINING DISTRIBUTED LAG

B. J. BALL

Department of Electrical Engineering
Mississippi State University, State College, Mississippi

ABSTRACT

When a closed loop system containing distributed lag, characterized by $e^{-\sqrt{s}T}$, in cascade with lumped parameter elements is analyzed using root locus techniques, two branches of the closed loop root loci originate at each simple pole of $GH(s)$ and terminate at each simple zero of $GH(s)$ as the system gain is increased from zero to infinity. This is contrary to the usual occurrence where one branch of the root loci originates at each pole of $GH(s)$ and migrates to a zero of $GH(s)$ as gain is increased to infinity.

In this paper, a Riemann surface of two connected sheets is used to display completely the root locus properties. The transformation $w = \sqrt{s}$ is then used to obtain a work surface of one sheet with the closed loop transfer function being a single-valued function of points on the surface. The nature of the roots on the two loci originating from a simple pole is examined, and they are shown to be terms in the denominator of the closed loop system of the form $(\sqrt{s} - a' - jb')$ rather than of the conventional form $(s - a - jb)$.

INTRODUCTION

The term $e^{-\sqrt{s}T}$, which characterizes the distributed lag in a distributed parameter system, may appear in the transfer function characterizing a system when the system includes such elements as an electrical cable, heat conduction in a continuous medium, transmission of air through a long pneumatic line, or concentration of a material in a diffusible solid. When a closed loop system containing distributed lag in cascade with lumped parameter elements is analyzed using root locus techniques, two branches of the closed loop root loci originate at each simple pole of $GH(s)$ and terminate at each simple zero of $GH(s)$ as the system gain is increased from zero to infinity.

An example is presented here where distributed lag exists in cascade with a lumped parameter lag network. The example is analyzed to show that the roots on the two branches originating from a simple pole represent terms of the form $(\sqrt{s} - a' - jb')$ rather than of the conventional form $(s - a - jb)$.

MAGNITUDE AND PHASE ANGLE LOCI

A specific example is selected as shown in Figure 1 for the purpose of plotting curves and evaluating roots. $T = 0.0153$ is used. However, each equation is presented in general terms first and then values substituted.

To facilitate the separation of the transfer function into an attenuation and phase angle for establishing the loci of roots shown in Figure 2 let

$$s = \sigma + j\omega \quad (1)$$

Then

$$\sqrt{sT} = \sqrt{T} \sqrt{\sigma + j\omega} \quad (2)$$

The n roots of a complex number, Z , are given by

$$Z_0 = \sqrt[n]{r} e^{j\left(\frac{\theta+2k\pi}{n}\right)} = \sqrt[n]{r} \left[\cos\left(\frac{\theta+2k\pi}{n}\right) + j \sin\left(\frac{\theta+2k\pi}{n}\right) \right] \quad (3)$$

where $k = 0, 1, \dots, n-1$ and $Z_0 = \sqrt[n]{Z}$

The two roots of the exponent, $\sqrt{sT}$, are then¹

for $k = 0$

$$\sqrt{sT} = \sqrt{\frac{T}{2}} (\sqrt{\sigma^2 + \omega^2} + \sigma)^{1/2} + j\sqrt{\frac{T}{2}} (\sqrt{\sigma^2 + \omega^2} - \sigma)^{1/2} \quad (4)$$

and for $k = 1$

$$\sqrt{sT} = -\sqrt{\frac{T}{2}} (\sqrt{\sigma^2 + \omega^2} + \sigma)^{1/2} - j\sqrt{\frac{T}{2}} (\sqrt{\sigma^2 + \omega^2} - \sigma)^{1/2} \quad (5)$$

The transfer function

$$GH(s) = e^{-\sqrt{sT}} \frac{(s+A)}{(s+B)} \quad (6)$$

then has two branches, one for each of the two roots of the exponent, $\sqrt{sT}$ which are

$$GH(s) = e^{-[T/2(\sqrt{\sigma^2 + \omega^2} + \sigma)]^{1/2}} e^{-j[T/2(\sqrt{\sigma^2 + \omega^2} - \sigma)]^{1/2}} \frac{(\sigma + j\omega + A)}{(\sigma + j\omega + B)} \quad (7)$$

and

$$GH(s) = e^{[T/2(\sqrt{\sigma^2 + \omega^2} + \sigma)]^{1/2}} e^{j[T/2(\sqrt{\sigma^2 + \omega^2} - \sigma)]^{1/2}} \frac{(\sigma + j\omega + A)}{(\sigma + j\omega + B)} \quad (8)$$

Note that these branches of the transfer function are the same except for the sign of the exponent. Using the numerical values stated above, the phase contribution and attenuation contribution from the simple lag network shown in the bracket of equation (7) and (8) is the same in each equation and is shown on the right hand side of Figures 3 and 4 respectively. Lines of constant-phase contribution from the exponential term of the branch described by equation (7) are shown in Figure 3 as $-\pi, -3\pi$ lines with additional negative phase contribution farther to the left, and those of the branch described by equation (8) are shown in Figure 3 as $\pi, 3\pi$ lines. Constant attenuation lines for the exponential term of the branch described by equation (7) appear in Figure 4 as $+20, +40, +60, +80$, etc. DB in the upper half-plane, and as $-20, -40, -60, -80$, etc. DB in the lower half-plane while for the branch described by equation (8) they appear as $-20, -40, -60, -80$, etc. DB in the upper half-plane and as $20, 40, 60, 80$, etc. DB in the lower half-plane. The phase contribution from the exponential terms and the simple lag terms combine for the branch described by equation (7) to produce loci of $(-\pi - 2n\pi)$ for $n = 0, 1, 2, \dots$ of which two are shown in Figures 2 and 5. The locus from the pole to the zero in the upper half-plane is produced by this branch and so is the locus to the left of the zero shown dashed in the lower half-plane and solid in the upper half-plane. Other loci for this branch exist on to the left which start in the fourth quadrant at ∞ , cross the negative real axis and continue to ∞ in the first quadrant as $K \rightarrow \infty$. The branch described by equation (8) produces loci of $(+\pi + 2n\pi)$ for $n = 0, 1, 2, \dots$ two of which are shown in Figures 2 and 6. One is the locus from the pole to the zero in the lower

half-plane, and one is shown solid in the lower half-plane and dashed in the upper half-plane.

RIEMANN SURFACE REPRESENTATION

The s-plane presentation shown in Figure 2 is not convenient to work with since each point in the s-plane corresponds to two separate values of the transfer function. A Riemann surface of two connected sheets can be used to avoid this condition. Churchill² defines a Riemann surface as a generalization of the s-plane into a surface of more than one sheet such that a multiple-valued function has only one value corresponding to each point on that surface. Once such a surface is devised for a given function, that function is a single-valued function of points on the surface, and the theory of single-valued functions can be used in dealing with the function. The loci of Figure 2 are presented with the part resulting from the branch of the transfer function described by equation (7) shown as the upper sheet of the Riemann surface in Figure 5 and the part described by equation (8) shown as Figure 6. The checked edge, $\vee \vee \vee$, of Figure 5 is joined to the checked edge, $\wedge \wedge \wedge$, of Figure 6 to produce the complete Riemann surface of two sheets shown in Figure 7.

REPRESENTATION IN THE w-PLANE

The Riemann Surface of Figure 7 suggests the linear transformation

$$w = \sqrt{s} \quad (9)$$

to obtain a work space of one sheet and retain the condition that each point in the w-plane corresponds to one value of the transfer function. Under the transformation, constant-phase lines for the exponential terms of equation (7) and (8) become straight linearly spaced horizontal lines in the upper and lower half-plane respectively; and equal-attenuation lines become straight vertical lines.¹

Using this transformation the loci of the roots for the example system are shown in Figure 8.

NATURE OF THE CLOSED LOOP ROOTS

Roots of $[1+GH(s)]$ and of $[1+GH(\omega)]$ respectively for $K = 2$ are carried on loci in Figures 2, 5, 6, 7, and 8 as P_1 , P_2 , P_3 , and P_4 . From Figure 8 the closed loop transfer function is obtained as

$$\frac{C}{R}(w) = \frac{N(w)(w-j15)(w+j15)}{D(w)(w-4-j14.5)(w-4+j14.5)(w-7-j22.5)(w-7+j22.5)}$$

where $N(w)$ and $D(w)$ represent terms not shown in the area of Figure 8.

This yields the closed loop transfer function in the Laplace domain as

$$\frac{C}{R}(s) = \frac{N(s)(s+22.5)}{D(s)(\sqrt{s}-4-j14.5)(\sqrt{s}-4+j14.5)(\sqrt{s}-7-j22.5)(\sqrt{s}-7+j22.5)}$$

P_1 in Figure 2 represents a term of the form $(\sqrt{s}-4-j14.5)$ located at $s = -193+j15$ rather than a pole yielding the term $(s+193-j15)$. The same is the case for P_2 , P_3 , and P_4 .

CONCLUSIONS

It is concluded that the two loci which originate at a simple pole and terminate at a simple zero as $0 < K < \infty$ for systems which contain distributed lag in cascade with lumped-parameter networks yield terms along the loci which are of the form $(\sqrt{s} + a' \pm jb')$ rather than terms of the form $(s + a + jb)$. These terms combine for $K = 0$ or for $K = \infty$ to produce a pole of $\frac{C}{R}(s)$ at the pole of $GH(s)$ or at the zero of $GH(s)$ respectively.

NOTATION

A	=	A complex number
B	=	A complex number
a	=	Real part of a root
b	=	Imaginary part of a root
C(s)	=	System controlled variable or output
D	=	Denominator
G(s)	=	Transfer function, forward components
H(s)	=	Transfer function, feedback components
K	=	System gain, variable
k	=	An integer used as an index
N	=	Numerator
n	=	An integer used as an index
R(s)	=	System reference or input
r	=	Magnitude of s
s	=	Laplace transformation Parameter
T	=	Distributed-lag time constant
w	=	Transformation parameter, $w = \sqrt{s}$
Z, Z ₀	=	Complex numbers
σ	=	Real part of s
ω	=	Imaginary part of s

REFERENCES

1. Ball, B. J., and Rekoﬀ, M. G., Jr. "An Analysis of the Distributed Lag," ISA Transactions, Vol. 5, No. 2, April 1966, pp 146-155.
2. Churchill, R. V. Complex Variables and Applications. New York, N. Y.: McGraw-Hill Book Co., Inc., 1960, pp 273-280.

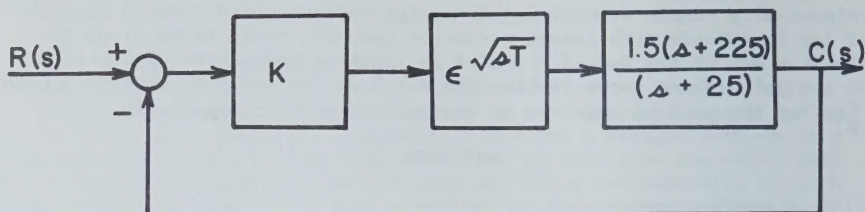


Figure 1. Block diagram of a closed loop system containing distributed lag in cascade with a lumped parameter lag network.

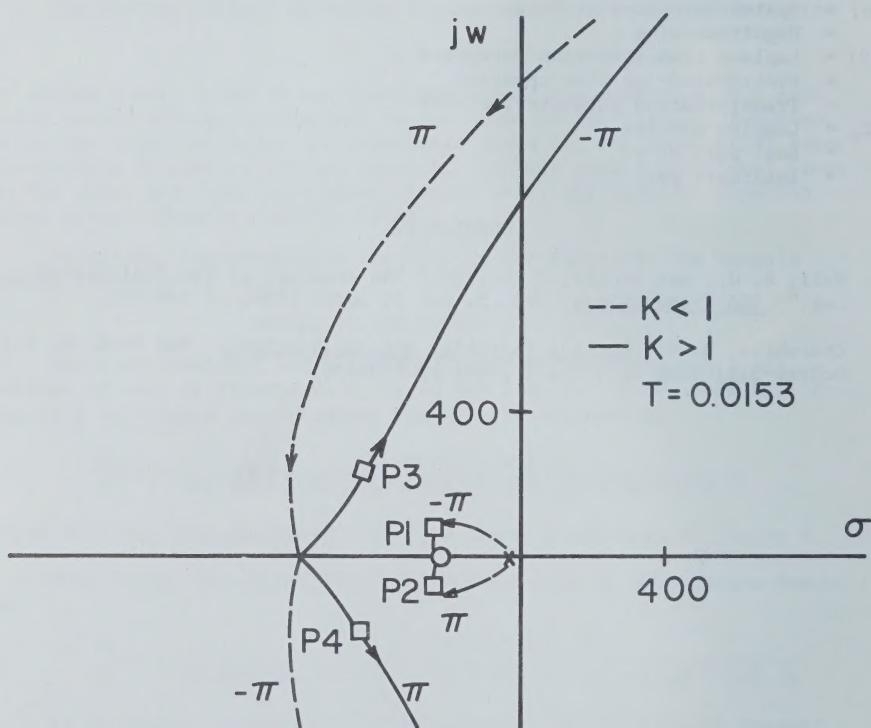


Figure 2. Four branches of the root loci in the s-plane of a system containing distributed lag and a lumped parameter lag network shown in Figure 1.

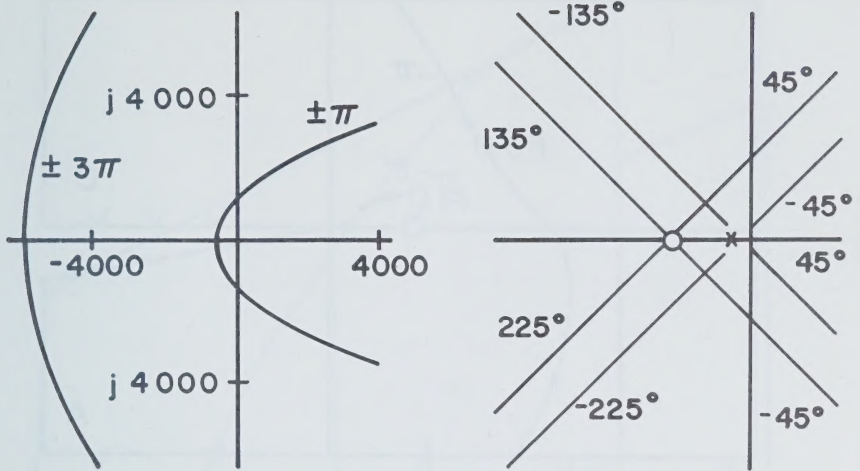


Figure 3. Loci of constant phase for (Left) $e^{-\sqrt{sT}}$ where $T = 0.0153$ and (Right) $(s + 225)/(s + 25)$.

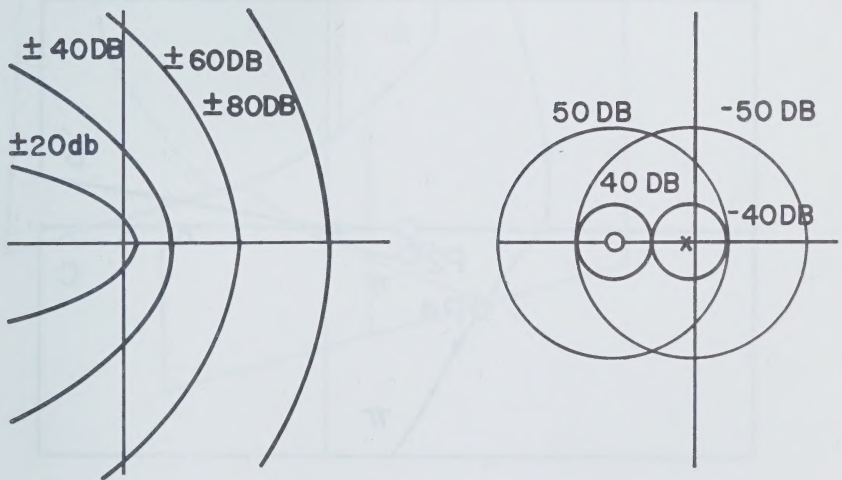


Figure 4. Loci of constant magnitude for (Left) $e^{-\sqrt{sT}}$ where $T = 0.0153$ and (Right) for $(s + 225)/(s + 25)$.

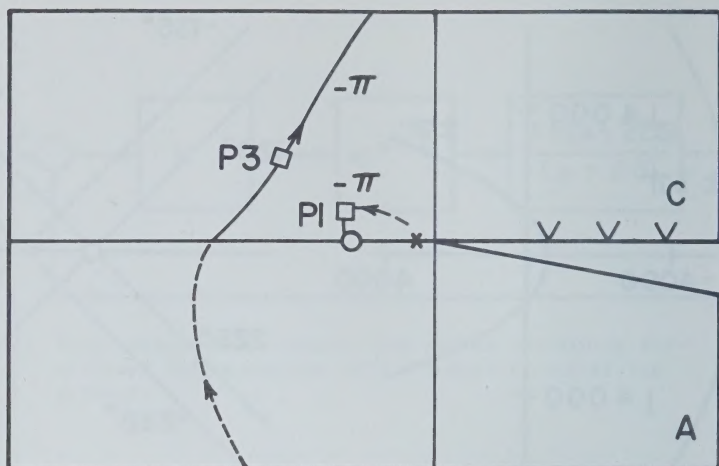


Figure 5. The upper one-half of a Riemann surface representation for the loci of the system of Figure 1.

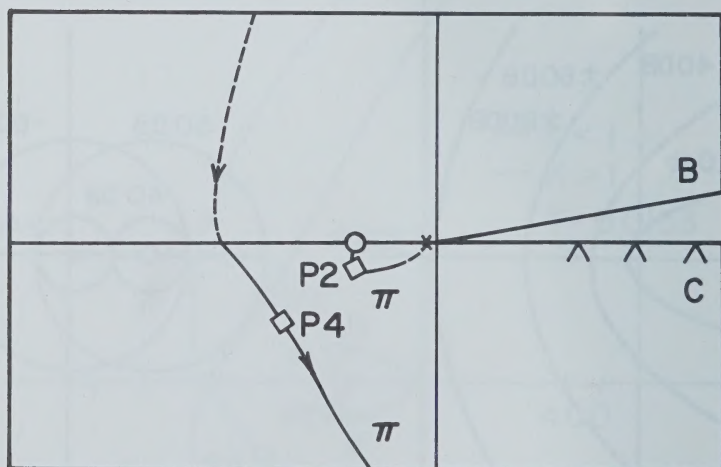


Figure 6. The lower one-half of a Riemann surface representation for the loci of the system of Figure 1.

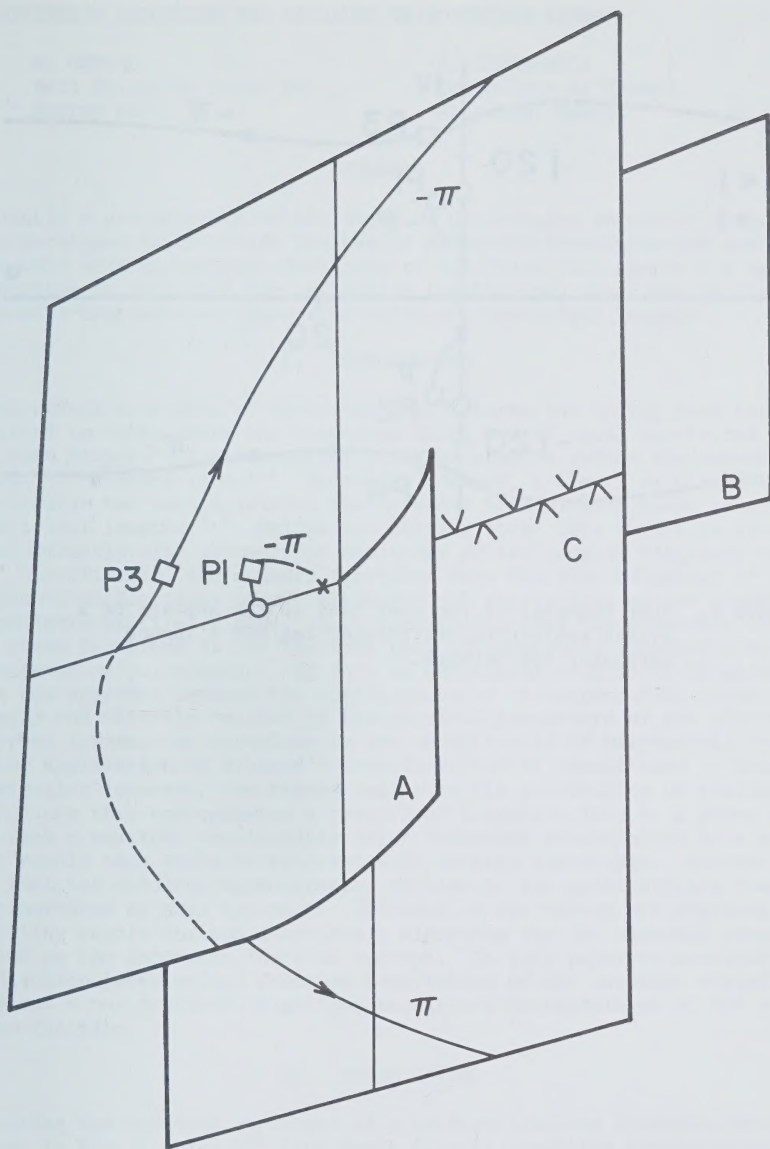


Figure 7. A Riemann surface representation for the loci of the system of Figure 1.

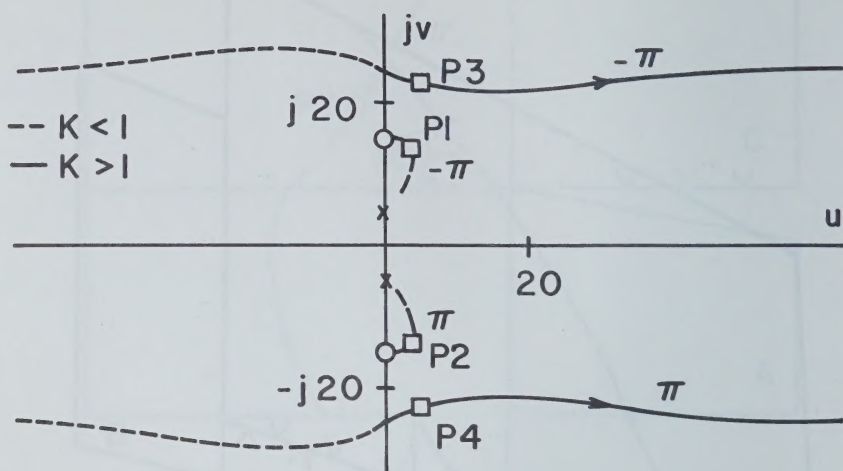


Figure 8. Four branches of the root loci in the w -plane of a system containing distributed lag and a lumped parameter lag network.

A SYNTHESIS ALGORITHM FOR CASCADED TRANSMISSION LINES

A. GERSHO
Bell Telephone Labs. Inc.,
Murray Hill, N. J.

B. KINARIWALA
University of Hawaii,
Honolulu, Hawaii

SUMMARY

Recently a new approach to the study of distributed parameter structures was developed which avoids the use of Richard's transformation and deals directly with meromorphic functions of s . Using this approach a synthesis algorithm is presented for reflection (scattering) functions in the form of cascaded transmission lines with arbitrary electrical lengths.

I. INTRODUCTION

Theoretical treatment of distributed structures has in the past been limited to cases where the component lines are of equal electrical length so that Richard's transformation could be used to reduce the network functions to rational form.¹⁻⁴ Recently, however, a theory of distributed structures has been developed for cascaded transmission lines of arbitrary electrical lengths.^{5,6} Unlike the earlier work, this theory is able to deal directly with meromorphic functions of the complex frequency variable s . Specifically, the network functions have the form of ratios of sums of exponential functions of s . Although such expressions appear somewhat formidable at first glance, it is interesting that the necessary treatment of these functions is for the most part of comparable simplicity to the treatment of polynomials. In fact an additional simplicity is gained from the new approach because the coefficients of the exponential terms are simply and directly related to the physical parameters of the structure, whereas information contained in the coefficients of polynomials obtained after application of Richard's transformation is camouflaged or lost. Of particular interest, the theory indicates the possibility of realizing a structure that approximates a prescribed insertion loss to a given accuracy in such a way that considerably fewer component transmission line sections may result than would be achieved with earlier techniques. Another feature is that the unsolved approximation problem of the multivariable theory¹⁰ is circumvented by this approach. Included in the theory are physical realizability conditions and a synthesis algorithm for the cascaded structure based on the impedance function concept. In this paper we consider the reflection (scattering) function description of the cascaded structure and present a new synthesis algorithm based on a decomposition of the reflection function.

II. FORMULATION

Consider the cascaded structure of n uniform lossless transmission lines as shown in Fig. 1. The k^{th} line has a (finite) positive characteristic impedance R_k and an electrical length τ_k , i.e., $R_k = \sqrt{L_k/C_k}$, and $\tau_k = \sqrt{L_k C_k} z_k$ where L_k and C_k are respectively the inductance and capacitance per unit length of the k^{th} line and z_k is the physical length of the k^{th} line. The structure is terminated in a load resistance R_{n+1} (or an infinite line with characteristic impedance R_{n+1}). A generator resistance R_0 is assumed present at the input end.

We define the reflection coefficient ρ_k for the k^{th} junction, i.e., the junction between the k^{th} and $k+1^{\text{th}}$ lines, according to

$$\rho_k = \frac{R_{k+1} - R_k}{R_{k+1} + R_k}, \quad k = 0, 1, \dots, n. \quad (1)$$

Since $0 < R_k < \infty$, it may be seen from (1) that

$$|\rho_k| < 1, \quad k = 0, 1, \dots, n-1 \quad (2)$$

and

$$|\rho_n| \leq 1 \quad (3)$$

with $|\rho_n| = 1$ when the termination is an open or short circuit. Without loss of generality we may assume that $\rho_k \neq 0$ for $k = 0, 1, 2, \dots, n$.

In place of the ordinary voltage and current variables, it is appropriate for our purposes to characterize the signals travelling through the structure using the reflected and incident voltages familiar from scattering matrix theory.⁷ If V_k and I_k are the (Laplace transformed) voltage and current at the k^{th} junction, then the reflected voltage X_k and incident voltage Y_k are defined by

$$X_k = V_k - R_{k+1}I_k; \quad Y_k = V_k + R_{k+1}I_k \quad (4)$$

The reflection (scattering) function $\Gamma_k(s)$, at the k^{th} junction is then defined by

$$\Gamma_k(s) = \frac{X_k}{Y_k} \quad (5)$$

For $k = 0$, (5) defines $\Gamma_0(s)$, the input reflection function of the structure. It follows from (4) that the input impedance function of the structure, $Z_0(s) = V_0/I_0$, is related to $\Gamma_0(s)$ according to

$$\Gamma_0(s) = \frac{Z_0(s) - R_0}{Z_0(s) + R_0} \quad (6)$$

By applying the change of variable (4) to the standard transmission line equations,⁸ the relation between the signal variables X, Y at the two ends of the kth transmission line is found to be

$$X_{k-1} = \rho_{k-1} e^{2s\tau_k} Y_k + X_k, \quad (7)$$

$$Y_{k-1} = e^{2s\tau_k} Y_k + \rho_{k-1} X_k, \quad (8)$$

from which, using (5), we obtain

$$\Gamma_{k-1}(s) = \frac{\rho_{k-1} e^{2s\tau_k} + \Gamma_k}{e^{2s\tau_k} + \rho_{k-1} \Gamma_k} \quad (9)$$

Also, we have

$$\Gamma_n(s) = \rho_n \quad (10)$$

which follows from (4) and (5). Equations (9) and (10) together provide the complete inductive rule from which $\Gamma_0(s)$ can be uniquely determined for a particular structure.

Certain key properties of $\Gamma_k(s)$ are readily deduced from (9) and (10). In particular, it may be seen that $\Gamma_k(s)$ can always be expressed in the form

$$\Gamma_k(s) = \frac{A_k(s)}{B_k(s)} = \frac{\sum_{r=1}^m a_{kr} e^{su_r}}{\sum_{r=1}^m b_{kr} e^{su_r}} \quad (11)$$

with u_r real, nonnegative, increasing with r , $u_1 = 0$, and m depends on k . Furthermore

$$\frac{a_{km}}{b_{km}} = \frac{b_{kl}}{a_{kl}} = \rho_k \quad (12)$$

from which it follows that a nonzero constant term is always present in A_k and B_k and that the highest exponent term e^{sum} is also present in both A_k and B_k . Exponential sums such as $A_k(s)$ and $B_k(s)$ in distributed network theory play a role equivalent to the role of polynomials in lumped network theory.

III. SOME PREVIOUS RESULTS^{5,6} AND THEIR IMPLICATIONS

The insertion loss function $L(\omega)$ of a loaded two port structure is defined as the ratio of the power available from a sinusoidal generator of frequency ω to the power delivered to the load. Using Fourier series techniques, a prescribed insertion loss function can be approximated by a sum of cosine terms. The possibility of finding a cascaded transmission line structure having such an insertion loss function has been definitively answered by the Insertion Loss Realizability Theorem⁶: An insertion loss $L(\omega)$ is physically realizable as the cascaded structure of Fig. 1 if and only if it has the form

$$L(\omega) = \sum_{k=1}^n c_k \cos \omega v_k, \quad c_k, v_k \text{ real},$$

and it satisfies the passivity requirement $L(\omega) \geq 1$. An insertion loss function satisfying these conditions may be synthesized in the following way. From

$$L(-js) = \sum_0^m c_k (e^{sd_k} + e^{-sd_k})$$

it is possible to obtain an input reflection function $\Gamma_0(s)$ which satisfies

$$\Gamma_0(s)\Gamma_0(-s) = 1 - \frac{1}{L(-js)}$$

in which $\Gamma_0(s)$ is a ratio of exponential sums of the form (10), and is analytic in the entire right-half s plane. (An algorithm to test the analyticity of such functions is available.⁹) Choosing R_0 arbitrarily, the driving point impedance Z_0 is then obtained from

$$Z_0(s) = R_0 \frac{1 + \Gamma_0(s)}{1 - \Gamma_0(s)} \quad (13)$$

which is the inverse of relation (6). An impedance function obtained in this manner will always satisfy the following necessary and sufficient conditions for realizability with the cascaded structure: (a) $Z_0(s)$ must be a positive-real function of s , and (b) $\text{Ev } Z_0(s) = c/D(s)$ where $D(s)$ is an exponential sum and $c \geq 0$. In the case $c = 0$ the structure is lossless, i.e., $R_{n+1} = 0$ or ∞ . A decomposition algorithm determines successively the parameters R_1 and τ_1 , R_2 and τ_2 , etc. This characterizes a (nonunique) structure having the specified insertion loss $L(\omega)$ and the (unique) cascaded structure having the input impedance function $Z_0(s)$.

The above synthesis procedure automatically yields a realizable reflection function, $\Gamma_0(s)$, from the given realizable insertion loss $L(\omega)$. Instead of using (13) to obtain $Z_0(s)$, and then applying the impedance decomposition algorithm, the algorithm which we present in the following section may be applied directly to the reflection function yielding the parameter values of the structure. The explicit conditions for physical realizability of a reflection function $\Gamma_0(s)$ can be easily deduced from the given realizability conditions for $Z_0(s)$ with the help of the relations (6) and (13). The necessary and sufficient realizability conditions for a ratio of exponential sums $\Gamma_0(s) = A_0(s)/B_0(s)$ of the form (11) to be realizable as the reflection function of a cascaded transmission line structure are: (a) $|\Gamma(s)| \leq 1$ in the right half and on the imaginary axis of the s plane, and (b) $B_0(s)B_0(-s) - A_0(s)A_0(-s) = c$ where c is a nonnegative constant.

The realizability conditions for $Z_0(s)$ or $\Gamma_0(s)$ in the case of a lossless structure (where $\text{Ev } Z_0(s) = 0$ and $\Gamma_0(s)\Gamma_0(-s) = 1$) reduce to the necessary passivity condition for the driving point impedance or reflection function of any passive distributed structure consisting of uniform lossless transmission lines. This implies that the cascaded structure terminated in a short or open circuit is a canonical realization for the impedance or reflection function of any passive distributed structure of uniform lossless lines.

IV. STATEMENT OF THE ALGORITHM

Given a realizable reflection function

$$\Gamma_0(s) = \frac{\sum_{k=1}^m a_k e^{su_k}}{\sum_{k=1}^m b_k e^{su_k}}$$

with u_k nonnegative and increasing with k ,

(1) Identify

$$\frac{a_m}{b_m} = \rho_0$$

the reflection coefficient of the first junction,

(2) Letting

$$a'_k = a_k - \rho_0 b_k$$

$$b'_k = b_k - \rho_0 a_k$$

form the function

$$\Gamma'_0(s) = \frac{\sum_{k=1}^m a'_k e^{s u_k}}{\sum_{k=1}^m b'_k e^{s u_k}}$$

(3) Identify the lowest u_k with nonzero b'_k as $2\tau_1$, giving the electrical length of the first line.

(4) Obtain

$$\Gamma_1(s) = \frac{\sum_{k=1}^m a'_k e^{s u_k}}{\sum_{k=1}^m b'_k e^{s(u_k - 2\tau_1)}}$$

These steps are then repeated using $\Gamma_1(s)$ to obtain the next transmission line section and so on until the process terminates to a constant which will be the reflection coefficient ρ_n . If the characteristic impedances are needed, they may be obtained by choosing R_0 arbitrarily and letting

$$R_{k+1} = R_k \frac{1 + \rho_k}{1 - \rho_k}$$

for $k = 0, 1, 2, \dots, n$, which is the inverse of relation (1).

V. PROOF OF THE ALGORITHM

Using the iterative relation (9) and expressing Γ_k in the form A_k/B_k we obtain

$$\Gamma_0(s) = \frac{A_0(s)}{B_0(s)} = \frac{\rho_0 e^{2s\tau_1} B_1(s) + A_1(s)}{e^{2s\tau_1} B_1(s) + \rho_0 A_1(s)} \quad (14)$$

where A and B are exponential sums as in (11). From (12) it is clear that the first step of the algorithm will correctly identify ρ_0 . The second step yields the function $\Gamma'_0(s)$ according to

$$\Gamma'_0(s) = \frac{A_0(s) - \rho_0 B_0(s)}{B_0(s) - \rho_0 A_0(s)}$$

Applying this to (14) shows that

$$\Gamma_0(s) = \frac{(1-\rho_0^2)A_1(s)}{(1-\rho_0^2)B_1(s)e^{2s\tau_1}}$$

Because the constant term in $B_1(s)$ cannot be zero, step (3) of the algorithm must correctly identify $2\tau_1$. Consequently step (4) yields

$$\Gamma_1(s) = \frac{\alpha A_1(s)}{\alpha B_1(s)}$$

with

$$\alpha = 1 - \rho_0^2 \neq 0.$$

From the iterative relations (9) and (10) it is clear that repetition of the algorithm correctly identifies successive line sections in the same way, without being affected by the nonzero constant factor in numerator and denominator. The process terminates with

$$\Gamma_n(s) = \frac{\beta \rho_n}{\beta} = \rho_n$$

where

$$\beta = \prod_{k=0}^{n-1} (1 - \rho_k^2) \neq 0.$$

This completes the proof.

The algorithm as presented here is in a general form which can handle certain degeneracies that may occur. In the nondegenerate case, where there are no partial sums of electrical lengths τ_i equal to other partial sums of the electrical lengths, the algorithm becomes simpler and can be performed by inspection because no arithmetical operations on the coefficient are needed. This may be seen by noting that for each k either a'_k or b'_k must be zero in the nondegenerate case. In most cases when degeneracies do occur many of the coefficients a'_k and b'_k will still be zero.

Although to our knowledge no computer exists that can process irrational numbers, it is of academic interest to examine the case where the exponents u_k are not rational. In this event the possibility arises that the cascaded structure corresponding to the given reflection function has an infinite number of component lines so that the algorithm does not terminate in a finite number of iterations. The realizability conditions remain valid with the implicit understanding that the corresponding cascaded structure may have an infinite number of component lines but with finite total electrical length. Of course the given reflection function can be approximated as closely as desired by a reflection function with rational u_k resulting in a realization with a finite number of component lines.

VI. A NUMERICAL EXAMPLE

$$\Gamma_0(s) = \frac{.5e^{28s} - .015e^{22s} - .03e^{20s} + .21e^{14s} - .006e^{8s} - .3e^{6s} + .10}{e^{28s} - .03e^{22s} - .06e^{20s} + .12e^{14s} - .003e^{8s} - .15e^{6s} + .05}$$

$$\rho_0 = .5$$

$$\Gamma_1(s) = \frac{.15e^{14s} - .0045e^{8s} - .225e^{6s} + .075}{.75e^{14s} - .0225e^{8s} - .045e^{6s} + .015}$$

$$\tau_1 = 7, \rho_1 = .2$$

$$\Gamma_2(s) = \frac{-.216e^{6s} + .0720}{.720e^{6s} - .0216}$$

$$\tau_2 = 4, \rho_2 = -.3$$

$$\Gamma_3(s) = \frac{.06552}{.65520}$$

$$\tau_3 = 3, \rho_3 = .1$$

Thus, the structure consists of three transmission line sections resistively terminated with electrical lengths and reflection coefficients as given. Apart from an arbitrary impedance scaling factor, the structure is uniquely determined.

VII. CONCLUSIONS

The cascaded structure with arbitrary electrical lengths may be conveniently applied to such applications as microwave filters and multilayer thin-film optical filters. The approach of dealing directly with ratios of exponential sums has led to a simple synthesis algorithm for obtaining the cascaded structure with arbitrary electrical lengths from a prescribed reflection function. The algorithm may be conveniently incorporated into the practically important synthesis procedure for obtaining a cascaded structure having a prescribed insertion loss.

VIII. REFERENCES

1. P. I. Richards, "Resistor-Transmission-Line Circuits," Proc. IRE, Vol. 26, pp. 217-220, February 1948.
2. H. Ozaki and J. Ishii, "Synthesis of Transmission-Line Networks and the Design of UHF Filters," IRE Trans. on Circuit Theory, Vol. CT-2, pp. 325-336, December 1955.
3. H. J. Riblet, "General Synthesis of Quarter-Wave Impedance Transformers," IRE Trans. on Microwave Theory and Techniques, Vol. MTT-5, pp. 36-43, January 1957.
4. H. Seidel, "Synthesis of a Class of Microwave Filters," IRE Trans. on Microwave Theory and Techniques, Vol. MTT-5, pp. 107-114, April 1957.
5. B. K. Kinariwala, "Theory of Cascaded Structures: Lossless Transmission Lines," Bell Sys. Tech. J., Vol. 45, pp. 631-649, April 1966.
6. B. K. Kinariwala, "Theory of Cascaded Transmission Lines," Proc Symp. Generalized Networks, Polytechnic Institute of Brooklyn, 1966, pp. 345-352.
7. H. J. Carlin, "The Scattering Matrix in Network Theory," IRE Trans. on Circuit Theory, Vol. CT-6, pp. 88- June 1956.
8. E. Weber, "Linear Transient Analysis," Vol. 2, John Wiley and Sons, Inc., New York, N. Y., 1956, Chapter 6.
9. B. K. Kinariwala and A. Gersho, "A Stability Criterion for Nonuniformly Sampled and Distributed Parameter Systems," Bell Sys. Tech. J., Vol. 45 pp. 1153-1155, September, 1966.
10. T. Koga, "Synthesis of Finite Passive n-Ports with Prescribed Positive Real Matrices of Several Variables, IEEE Trans. on Circuit Theory, Vol. CT-15, No. 1, pp. 2-23, March 1968.

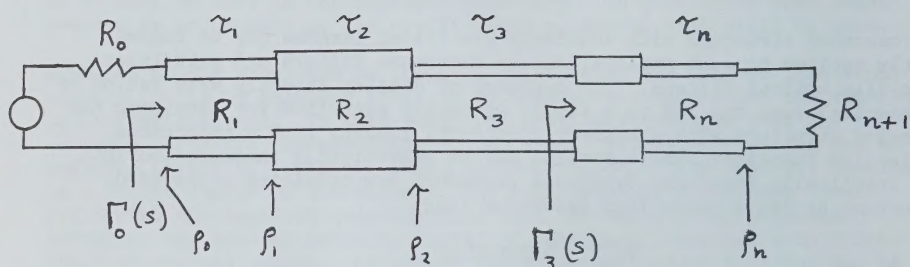


FIG. 1 CASCADED LINE STRUCTURE

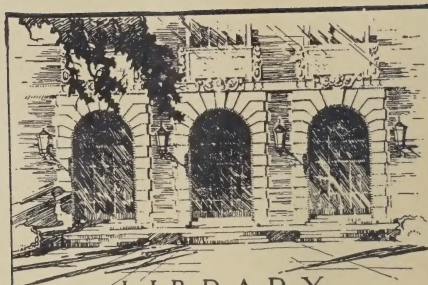
INDEX OF AUTHORS

AHMED, N.	133	CIRIC, V. V.	578
ALLEN, R. H.	311	CITRON, S. J.	68
ALTHAUS, H. L.	549	COFFMAN, E. G.	515
AROZULLAH, M.	131	COOK, G.	296
ATHANS, M.	42,661	CUIKAY, R. S.	274
BALL, B. J.	880	CURRY, R. E.	418
BECKER, S. T.	456	DAVIDSON, E. S.	538
BEDROSIAN, S. D.	406	DeDOES, D. H.	296
BERLEKAMP, E. R.	161	DELANSKY, J. F.	262
BLANK, G. L.	680	DEPALMS, G. F.	722
BONNELL, R. D.	619	DICKERSON, S. L.	619
BOSE, N. K.	256	DEO, N.	832
BOURQUIN, J. J.	870	DESOER, C. A.	852
BRANDEBERRY, J. E.	442	DOLEZAL, V.	305
BRANIN, F. H., Jr.	276	DUNN, W. R., Jr.	822
BRESHEARS, G. C.	31	DWYER, S. J., III	800
BRUCKNER, J.	475	EDDY, T. W.	189
CALAHAN, D. A.	321	EGER, T. E.	783
CAMERON, A. V.	426	ELTIMSAHY, A. H.	78
CHAN, S. P.	822	EPLEY, D. L.	523
CHANG, G.	800	FAY, J.	274
CHEN, Y-C. E.	523	FERGUSON, J. D.	671
CHIEN, Y. T.	792	FRIEND, R. C.	88
CHUNG, W. K.	568	FU, K. S.	10
CHYUNG, D. H.	465	GEFFE, P. R.	181
CILETTI, M. D.	296,654	GERSHO, A.	646,889

GLINSKI, G. S.	568	LEVINE, W. S.	661
GOKNAR, I. C.	232	LI, T. J.	10
GOPINATH, B.	207	LIBERTY, S. R.	408
GOURISHANKAR, V.	133	LIU, B.	219
GRAMMATICOS, A. J.	198	LIU, C. N.	30
HAYES, J. P.	732	LIU, R.	632
HEIZER, K. W.	862	LOCHRIE, W. D.	432
HEMAMI, H.	142	LOPRESTI, P. V.	456
HENDRICKS, C. L.	68	LUH, J. Y. S.	454
HERMAN, F. H.	406	LUPO, F. J.	247
HOLTZMAN, J. M.	589	McLAREN, R. W.	31
ISAACS, D.	432	McNAMEE, L. P.	396
KANEKO, T.	219	MAGUIRAGA, M.	454
KASHYAP, R. L.	1	MANCUSO, T. G.	621
KAZDA, L. F.	78	MARSHALL, T. G., Jr.	374
KELLY, G. L.	559,843	MARTENS, G. O.	237
KHATRI, H. C.	496	MARTENS, H. R.	228
KINARIWALA, B.	889	MARTIN, R. D.	773
KOIVO, A. J.	68	MASSEY, J. L.	153
KOMO, J. J.	800	MEDANIC, J.	702
KOZEMCHAK, E. B.	386	MEIER, L.	615
KREER, J. B.	52	METZE, G.	538,711,752
KURZ, L.	97	METZNER, J. J.	802
KUSIC, G. L.	485	MEYERHOFF, D. E.	396
LEAKE, R. J.	468,632	MICHEL, A. N.	860
LEE, S. C.	621	MICKLE, M. H.	62
LEEDS, J. V.	578	MIN, P. J.	10
LEININGER, G.	228	MOORE, D. J. H.	254
LESKOVAR, B.	122	MORGAN, B. S., Jr.	296

MORGAN, K. C.	802	SCHOOLEY, L. C.	559
MOWLE, F. J.	532	SCHUMACHER, P. J., Jr.	680
MUNTZ, R. R.	515	SCHUMER, M. A.	812
MURPHY, G. J.	690	SCHWARTZ, S. C.	773
MURRAY-LASSO, M. A.	386	SEMME LHACK, H. P.	88
MUSA, S. A.	151	SHARPE, G. E.	254
NIKOLIC, Z. J.	20	SILVER, H. I.	97
PAN, A. C.	31	SINHA, N. K.	700
PERKINS, W. R.	598	SO, H. C.	332,362
PERLIS, H. J.	505	SPRAQUE, R. E.	274
POTTLE, C.	311	STEIGLITZ, K.	814
POWELL, T. J.	761	STERN, T. E.	845
REDDY, D. C.	617	SUNAHARA, Y.	444
REKASIUS, Z. V.	671	SWONGER, C. W.	88
RESH, J. A.	232	TARASSOV, V. J.	505
RICHARDSON, M. H.	468	TEMPLE, V.	700
RICKER, D. W.	217	THOMAS, J. B.	634
RUGH, W. J.	690	TZENG, J. S.	870
SABLATASH, M.	264	UHRAN, J. J., Jr.	112
SAEKS, R.	230	VOGT, W. G.	62
SAIN, M. K.	153,408	WANG, K. C.	743
SANATHANAN, C. K.	617	WANG, K. U.	841
SANDBERG, I. W.	107,110,332	WEBB, R. P.	619
SANKARAN, S. V.	151	WEINER, P.	814
SARIDIS, G. N.	217	WIGGERT, D.	171
SATHE, S. T.	845	WILKIE, D. F.	598
SCHAEFER, J. F.	296	WILLHOFF, G. S.	88
SCHERTZ, D. R.	752	WILLSON, A. N., Jr.	107,110
SCHMIDLIN, D. J.	247	WOJCIK, A. S.	711

WOLF, J. K.	189
WU, M. Y.	852
WU, S. H.	442,475
WYNDRUM, R. W., Jr.	286
YAU, S. S.	722
YUAN, L. S.	52



LIBRARY
OF THE
UNIVERSITY
OF ILLINOIS

NOTICE: According to Sec. 19
(a) of the University Statutes,
all books and other library
materials acquired in any man-
ner by the University belong to
the University Library. When
this item is no longer needed
by the department, it should
be returned to the Acquisition
Department, University Library.

DIGITAL COMPUTER
LABORATORY
LIBRARY

DIGITAL COMPUTER
LABORATORY
LIBRARY

UNIVERSITY OF ILLINOIS-URBANA

621.31921AL54P C003
PROCEEDINGS. \$URBANA ILL
6 1968



3 0112 006279639